UOW High Performance Computing Cluster User's Guide

Information Management & Technology Services

University of Wollongong

(Last updated on March 31, 2015)

Contents

1. Overview **6**
- 1.1. Specification . . . 6
- 1.2. Access . . . 7
- 1.3. File System . . . 7

2. Quick Start **9**
- 2.1. Access the HPC Cluster . . . 9
  - 2.1.1. From within the UOW campus . . . 9
  - 2.1.2. From the outside of UOW campus . . . 9
- 2.2. Work at the HPC Cluster . . . 10
  - 2.2.1. Being familiar with the environment . . . 10
  - 2.2.2. Setup the working space . . . 11
  - 2.2.3. Initialize the computational task . . . 11
  - 2.2.4. Submit your job and check the results . . . 14

3. Software **17**
- 3.1. Software Installation . . . 17
- 3.2. Software Environment . . . 17
- 3.3. Software List . . . 19

4. Queue System **22**
- 4.1. Queue Structure . . . 22
  - 4.1.1. Normal queue . . . 22
  - 4.1.2. Special queues . . . 23
  - 4.1.3. Schedule policy . . . 23
- 4.2. Job Management . . . 23
  - 4.2.1. PBS options . . . 23
  - 4.2.2. Submit a batch job . . . 25
  - 4.2.3. Check the job/queue status . . . 27
  - 4.2.4. Submit an interactive job . . . 29
  - 4.2.5. Submit workflow jobs . . . 30
  - 4.2.6. Delete jobs . . . 31

5. Utilization Agreement **32**
- 5.1. Policy . . . 32
- 5.2. Acknowledgements . . . 32
- 5.3. Contact Information . . . 33

Appendices **35**

A. Access the HPC cluster from Windows clients **35**
A.1. Putty . . . 35
A.2. Configure 'Putty' with UOW proxy . . . 36
A.3. SSH Secure Shell Client . . . 38

B. Enable Linux GUI applications using Xming **41**
B.1. Install Xming . . . 41
B.2. Configure 'Putty' with 'Xming' . . . 42
B.3. Configure 'SSH Secure Shell Client' with 'Xming' . . . 44

C. Transfer data between the Windows client and the HPC cluster **46**
C.1. WinSCP . . . 46
C.2. SSH Secure File Transfer Client . . . 48

D. Transfer data at home (off the Campus) **49**
D.1. Windows OS . . . 49
D.2. Linux or Mac OS . . . 49
D.2.1. Transfer data from your home computer to the HPC cluster . . . 49
D.2.2. Transfer data from the HPC cluster to your home computer . . . 50

E. Selected Linux Commands **51**

F. Software Guide **53**
F.1. Parallel Programming Libraries/Tools . . . 53
F.1.1. Intel MPI . . . 53
F.1.2. MPICH . . . 53
F.1.3. OpenMPI . . . 54
F.2. Compilers & Building Tools . . . 55
F.2.1. CMake . . . 55
F.2.2. GNU Compiler Collection (GCC) . . . 56
F.2.3. Intel C, C++&Fortran Compiler . . . 56
F.2.4. Open64 . . . 57
F.2.5. PGI Fortran/C/C++ Compiler . . . 57
F.3. Scripting Languages . . . 58
F.3.1. IPython . . . 58
F.3.2. Java . . . 59
F.3.3. Perl . . . 59
F.3.4. Python . . . 59
F.4. Code Development Utilities . . . 60
F.4.1. Eclipse for Parallel Application Developers . . . 60
F.5. Math Libraries . . . 60
F.5.1. AMD Core Math Library (ACML) . . . 60
F.5.2. Automatically Tuned Linear Algebra Software (ATLAS) . . . 61
F.5.3. Basic Linear Algebra Communication Subprograms (BLACS) . . . 62
F.5.4. Basic Linear Algebra Subroutines (BLAS) . . . 62
F.5.5. Boost . . . 63
F.5.6. FFTW . . . 64
F.5.7. The GNU Multiple Precision Arithmetic Library (GMP) . . . 65
F.5.8. The GNU Scientific Library (GSL) . . . 66

F.5.9. Intel Math Kernel Library (IMKL) . . . 67
F.5.10. Linear Algebra PACKage (LAPACK) . . . 67
F.5.11. Multiple-Precision Floating-point with correct Rounding(MPFR) 68
F.5.12. NumPy . . . 68
F.5.13. Scalable LAPACK (ScaLAPACK) . . . 69
F.5.14. SciPy . . . 70
F.6. Debuggers, Profilers and Simulators . . . 71
F.6.1. HPCToolkit . . . 71
F.6.2. TAU . . . 71
F.6.3. Valgrind . . . 72
F.7. Visualization . . . 72
F.7.1. GNUPlot . . . 72
F.7.2. IDL . . . 73
F.7.3. matplotlib . . . 73
F.7.4. The NCAR Command Language (NCL) . . . 74
F.7.5. Ncview . . . 75
F.7.6. OpenCV . . . 75
F.8. Statistics and Mathematics Environments . . . 76
F.8.1. R . . . 76
F.9. Computational Physics and Chemistry . . . 78
F.9.1. ABINIT . . . 78
F.9.2. Atomic Simulation Environment (ASE) . . . 80
F.9.3. Atomistix ToolKit (ATK) . . . 81
F.9.4. AutoDock and AutoDock Vina . . . 81
F.9.5. CP2K . . . 82
F.9.6. CPMD . . . 84
F.9.7. DOCK . . . 85
F.9.8. GAMESS . . . 87
F.9.9. GATE . . . 88
F.9.10. GAUSSIAN . . . 91
F.9.11. Geant . . . 92
F.9.12. GPAW . . . 93
F.9.13. GROMACS . . . 94
F.9.14. MGLTools . . . 95
F.9.15. MOLDEN . . . 96
F.9.16. NAMD . . . 97
F.9.17. NWChem . . . 98
F.9.18. OpenBabel . . . 99
F.9.19. ORCA . . . 100
F.9.20. Q-Chem . . . 101
F.9.21. Quantum ESPRESSO . . . 102
F.9.22. SIESTA . . . 104
F.9.23. VMD . . . 106
F.9.24. WIEN2K . . . 106
F.9.25. XCrySDen . . . 107
F.10. Informatics . . . 107
F.10.1. Caffe . . . 107
F.10.2. netCDF . . . 108

F.10.3. netCDF Operator (NCO) . . . 109
F.10.4. RepastHPC . . . 110
F.10.5. SUMO(Simulation of Urban Mobility) . . . 110
F.11. Engineering . . . 111
F.11.1. MATLAB . . . 111
F.11.2. ANSYS,FLUENT,LSDYNA,ANSYS Electromagnetics Suite . . . 113
F.11.3. ABAQUS . . . 114
F.11.4. LAMMPS . . . 114
F.11.5. Materials Studio . . . 115
F.12. Biology . . . 115
F.12.1. ATSAS . . . 115
F.12.2. MrBayes . . . 116
F.12.3. PartitionFinder . . . 117
F.12.4. QIIME . . . 118

1. Overview

The UOW HPC cluster aims to provide computing services for the UOW academic staffs and postgraduate students in their research work at the University of Wollongong. The maintenance of the UOW HPC cluster is held by the Information Management & Technology Services (IMTS), UOW.

1.1. Specification

The UOW HPC cluster consists of 3 components:

- ***Login node*** (i.e. `hpc.its.uow.edu.au`) is used for users to login the HPC cluster. Users could use the login node to prepare jobs, develop and build codes, transfer data to and from their local storage locations. The login node is **NOT** used for job execution. Users **MUST** submit their jobs to the queue system rather than running the job at the login node directly.
- ***Compute nodes*** are the major computing infrastructures for executing jobs submitted by users. Users are not allowed to login any of the compute nodes directly.
- ***Storage servers*** provide major storage pool for users' home directory, job scratch directory and large data set storage. The whole storage pool is divided into different file systems with each for a specific purpose.

Table 1.1 shows the system details of each component.

Cluster Name	hpc.its.uow.edu.au
Compute Node Model	Dell PowerEdge C6145
Processor Model	Sixteen-Core 2.3 GHz AMD Opteron 6376
Processors per Node	4
Cores per Node	64
Memory per Node	256GB
Number of Nodes	22
Total Cores	1408
Total Memory	5,632GB
Network Connection	10 GbE
Operating System	CentOS 6.3
Queue System	Torque
Job Scheduler	Maui
Storage Capacity	120TB
Release Time	November 2013

Table 1.1.: The current UOW HPC cluster specifications

A list of software packages have been deployed at the HPC cluster spanning chemistry, physics, engineering, informatics, biology etc. Users can access these system-wide packages easily via software environment module package. For program development there are several compilers available such as Portland Group Workstation (PGI), GNU Compiler Collection (GCC), Open64 and Intel Cluster Studio XE. Several MPI libraries including OpenMPI, MPICH and Intel MPI are deployed to support parallel computing.

1.2. Access

UOW staffs are eligible to request their account at the HPC cluster. Students must get their supervisor to request an account at the HPC cluster. Contact HPC admin to apply for the account.

Once the account is enabled, users may use their normal IMTS supplied username and password to access the login node of the UOW HPC cluster via Secure Shell(SSH).

Users who are working within the campus can access the cluster by typing the following command in a Linux or Unix client:

```
ssh username@hpc.its.uow.edu.au
```

or by using the SSH tools such as Cygwin, Putty or SSH Secure Shell Client in a Windows desktop.

If working from the outside of UOW campus, users should firstly login the UOW gateway server `wumpus.uow.edu.au` and then `ssh` to the HPC cluster from there. UOW staff can login in the HPC cluster directly from the outside of UOW via VPN. Please note, the home directory on the HPC cluster is **NOT** the same as on other IMTS machines such as `wumpus`.

Note: The login node is ONLY for the job creation but not the job execution. User's job running at the login node may be terminated by the system automatically. Users MUST submit their jobs to the queue system by using the `qsub` command to run jobs at one or more compute nodes.

1.3. File System

- `/home/`*`username`* This is user's home directory. Most of user's activities should stay within it. It is globally accessible NFS from all nodes within the cluster. The quota of each user's home space is strictly set as 100GB. The current usage of the home directory can be viewed from the file `~/.usage`. If you need more storage space to run jobs, please contact the HPC administrator to request the extended storage space.
- `/extd/`*`username`* This directory is used to run jobs requesting more space than user's home directory. After sending the request to the HPC administrator and as soon as the request is approved, the requesters can access their extended space with the allocated quota. The current usage and quota of the extended space can also be viewed from the file `~/.usage`. Please note, the extended space will be periodically cleaned up and all it's files untouched for over 3 months will be

automatically removed. It is user's responsibility to move the useful results and data to the long-term storage space such as the home directory etc.

- `/hpc/software` Used for installing system-wide software packages accessible by multiple users. User needs to obtain permission from IMTS to deploy software here.
- `/tmp` This is the local directory attached to each node to store the intermediate files from various system commands and programs. Since it has very limited space, please do **NOT** set `TMPDIR` environment variable of user's applications to `/tmp` or put scratch data in it.
- `/hpc/tmp` This is the directory for users to exchange files.
- `/export/scratch/common` This is the directory where the running jobs should store all scratch or temporary data. Users should create the sub-directory named with the job id, copy the input and auxiliary files to it, execute the application, copy all results back to user's working directory after the job termination and finally remove the sub-directory. Please refer to the job script template at the HPC cluster as below when preparing your own job script to do all above
  `/hpc/tmp/examples/common/runjob_local.sh`
- `/hpc/data` Reserved for special requirements of data storage.

Note: The HPC storage space is protected via RAID (so loss of a single disk does not lose data) but is NOT backed up in any way. It is the user's responsibility to ensure that any important data is backed up elsewhere. Contact IMTS if you need special storage requirements.
To effectively manage the storage of the cluster, the following file system policies are applied:

- **Email Notifications**
  The system will send the notification email to the user whose home or extended space usage nearly approaches the quota. The user's running jobs will also be suspended to avoid the unexpected job termination. The user should take action immediately to reduce the storage usage at the corresponding space to the lower level. The system will automatically resume those suspended jobs after the user's free space is recovered to the normal size. If there is no response made by the user in a fixed time, all suspended jobs will be deleted to release the resources for executing other users' jobs. Check the file `.usage` under the home directory to view the current storage usage, i.e. '`more ~/.usage`'.
- **Extended Space**
  If the home space is not large enough to host a user's running jobs, the user could apply for the additional storage space. After the request is approved, the user will be allocated the extended space under the `/extd`?partition of the cluster. Please note, there is **NO** backup on user's extended space. Such extended space will be cleaned periodically and all data untouched over 3 months will be automatically deleted. It is user's responsibility to move the useful data from the extended space to either the home directory or other long-term storage locations.

2. Quick Start

2.1. Access the HPC Cluster

The hostname of the UOW HPC cluster login node is 'hpc.its.uow.edu.au'. To access it, you will need an account at the HPC cluster which can be created by sending e-mail to HPC admin.
There are several ways to access the HPC cluster depending on which operating system of your client computer and the location where you are trying to login the cluster.

2.1.1. From within the UOW campus

- Linux or Mac OS client
  You can easily log into the HPC cluster by typing the following command

  ```
  ssh USERNAME@hpc.its.uow.edu.au
  ```

  Note: USERNAME in this document always means your own username.
  If you have intention to execute the GUI (Graphical User Interface) applications at the HPC cluster, please add the -X flag to login the cluster, i.e.

  ```
  ssh -X USERNAME@hpc.its.uow.edu.au
  ```

- Windows client
  - 'Putty' or 'SSH Secure Shell Client'
    Both are the GUI SSH clients to access the remote Linux server. If you want to execute the GUI applications at the HPC cluster, please install a X-Windows terminal emulator (X Server) at your Windows desktop computer, such as 'Xming'. Please refer to Appendix A for instructions on configuring and using 'Putty', 'SSH Secure Shell' and 'Xming'.
  - Cygwin
    Cygwin is a collection of tools which could provide a Linux look and feel environment for Windows. Get the 'Cygwin' installed at your Windows client. Open a Cygwin-terminal and then use 'ssh' command to log into the HPC cluster as shown in Fig. 2.1, i.e.

    ```
    ssh USERNAME@hpc.its.uow.edu.au
    ```

    You need to install Cygwin/X and then login the HPC cluster with the -X flag to run the GUI applications at the HPC cluster.

2.1.2. From the outside of UOW campus

The UOW HPC cluster is only visible to the UOW intranet. UOW staffs can access the HPC cluster directly from the outside of the campus via UOW VPN.

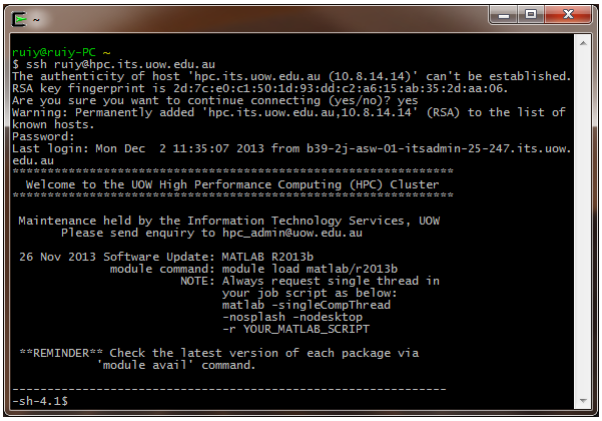
ruiy@ruiy-PC ~
$ ssh ruiy@hpc.its.uow.edu.au
The authenticity of host 'hpc.its.uow.edu.au (10.8.14.14)' can't be established.
RSA key fingerprint is 2d:7c:e0:c1:50:1d:93:dd:c2:a6:15:ab:35:2d:aa:06.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hpc.its.uow.edu.au,10.8.14.14' (RSA) to the list of known hosts.
Password:
Last login: Mon Dec 2 11:35:07 2013 from b39-2j-asw-01-itsadmin-25-247.its.uow.edu.au
Welcome to the UOW High Performance Computing (HPC) Cluster
Maintenance held by the Information Technology Services, UOW
Please send enquiry to hpc_admin@uow.edu.au
26 Nov 2013 Software Update: MATLAB R2013b
module command: module load matlab/r2013b
NOTE: Always request single thread in your job script as below:
matlab -singleCompThread -nosplash -nodesktop -r YOUR_MATLAB_SCRIPT
REMINDER Check the latest version of each package via 'module avail' command.
-sh-4.1$

Figure 2.1.: Use 'Cygwin' to login the HPC cluster. Username 'ruiy' is shown as an example.

Other users who are working from the outside of UOW need to login the `wumpus` gateway server firstly and then access the HPC cluster from there, i.e. type

```
ssh username@wumpus.uow.edu.au
```

from your home client and then type

```
ssh username@hpc.its.uow.edu.au
```

at the `wumpus` server.

2.2. Work at the HPC Cluster

As soon as you connect to the HPC cluster, you are entering the Linux OS. Linux has different commands from MS Windows or DOS and different ways of doing things. You will almost exclusively be presented with a text-based command line prompt at which you are expected to type commands - no point and click. Thus you need to be familiar with the Linux Command Line and you need to know what commands to type at the prompt. We will go through a basic set of Linux commands in this document which will allow you to start using the Linux operating system and conduct your HPC computations (summarized in Appendix E). Please refer to the advanced reference book for more detail usage of Linux commands. All the Linux commands are fully documented in an online manual, often referred to as the '`man pages`'. The manual can be invoked from the command line by typing '`man cmd`', where '`cmd`' is the command you wish to see the documentation for.

2.2.1. Being familiar with the environment

First log into the HPC cluster by using the methods mentioned in the preceding section. To view your current working directory, type '`pwd`' in the command line. You will be placed at your home directory, i.e. /home/*USERNAME* after each login. Every user has his/her own home directory which is inaccessible by other users. In most circumstances, users should keep the work within their own home directory.
To view a list of files in a directory, use the '`ls`' command. The '`ls`' command has many options and these options will dictate what output can be extracted. Please view the output from the command '`man ls`' to learn the details on these options.

2.2.2. Setup the working space

If you need to create different directories for each specific research topic, use the 'mkdir' command to create a directory named, for example, 'task', i.e.

```
mkdir task
```

To make sure the directory has been created, use the 'ls' command to list all directories and files under the present directory.

Now access your 'task' directory by typing:

```
cd task
```

'cd' command is used to enter a target directory. Type 'pwd' command to check whether you have moved to the 'task' directory.

2.2.3. Initialize the computational task

Generally, to start your computational task, you will need at least two files, i.e. the input file for the scientific program and the job script file to be submitted to the queue system.

Create input file for the scientific program

Normally a scientific program needs input files to run. For example, if you run a quantum chemistry computation, using Gaussian, NAMD, GAMESS etc, you have to prepare an input file containing information spanning the method, basis set, molecular geometry information and so on. There are two choices on preparing such input file: create the input file at the HPC cluster by using a Linux text editor, or create the input file at your desktop computer and then transfer it to the cluster.

1. Create the input file at the HPC cluster
   There are many popular Linux based text editors ready for use such as 'gedit', 'vi', 'emacs', 'nano' etc. (For some GUI text editors like 'gedit', you need to have a X-server installed at your client machine. Refer to Appendix B.1 for instructions on installing the X-server, 'Xming'). You could pick anyone up as your favorite editor. Let's consider a simple case to clarify the whole process.
   Supposing you are going to calculate the multiplication between the matrix A and B by using MATLAB package. You then need to create an input file which contains the definition of A and B, and the matrix operation command to calculate their multiplication.
   - If you have no X server installed on your desktop, you can use text-mode editors to create the input file. For example, you can use program 'nano' to create your input file. Type the following commands:

     ```
     cd ~/task
     nano matrix.m
     ```

     The character '~/' represents the user's home directory and '~/task' means the 'task' directory under the user's home directory at the HPC cluster. The above commands will first bring you to the 'task' directory and then open a file named 'matrix.m' in the 'nano' editor. Type the following contents in

the editor's window as shown in Fig. 2.2:

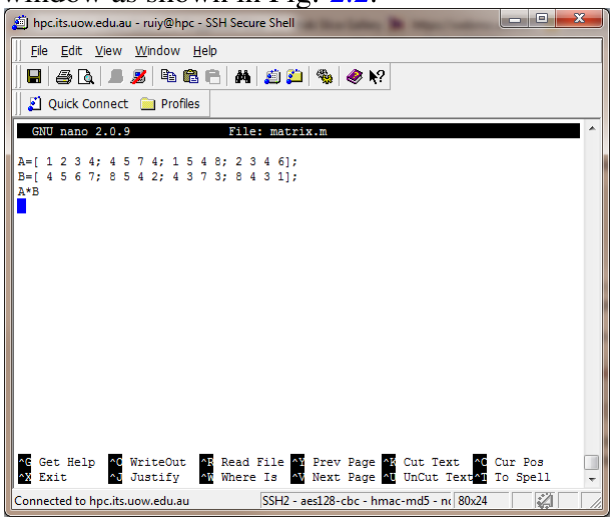
hpc.its.uow.edu.au - ruiy@hpc - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
GNU nano 2.0.9 File: matrix.m
A=[1 2 3 4; 4 5 7 4; 1 5 4 8; 2 3 4 6];
B=[4 5 6 7; 8 5 4 2; 4 3 7 3; 8 4 3 1];
A*B
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
Connected to hpc.its.uow.edu.au
SSH2 - aes128-cbc - hmac-md5 - n
80x24

Figure 2.2.: Use `nano` to create your input file.

Type 'Ctrl'+'X' to exit 'nano'. Note that operation keys are displayed at the bottom of 'nano' as shown in Fig. 2.2 and '^' means 'Ctrl' in your keyboard.

- If you have logged into the cluster with X server support then you can use program '`gedit`' to create such an input file named `matrix.m` under the directory '`~/task`':

```
cd ~/task
gedit matrix.m
```

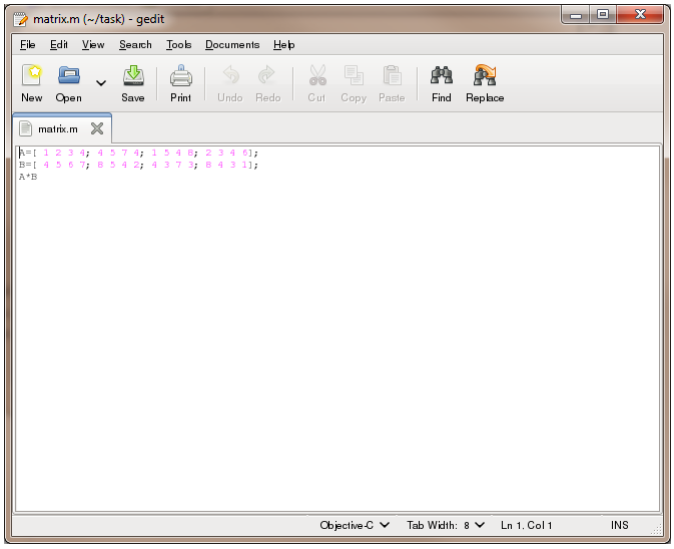
matrix.m (~/task) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace
matrix.m
A=[1 2 3 4; 4 5 7 4; 1 5 4 8; 2 3 4 6];
B=[4 5 6 7; 8 5 4 2; 4 3 7 3; 8 4 3 1];
A*B
Objective-C Tab Width: 8 Ln 1, Col 1 INS

Figure 2.3.: Use `gedit` to create your input file.

Type the same content as above in the 'gedit' window as shown in Fig. 2.3 and then Save & Quit. The first two lines specify two random 4×4 matrix's, A and B, and the third line is to calculate their matrix multiplication.

Make sure you have saved the contents to the file in both ways.

2. Transfer the input file to the cluster
   If you have created the above input file at your desktop computer then you could use the methods mentioned in Appendix C to copy it to the '`task`' directory of the HPC cluster. For example, in the same directory containing the input file at your Linux/Mac client computer, type

```
scp matrix.m  USERNAME@hpc.its.uow.edu.au:~/task
```

NOTE You can also solve the above jobs by using the GUI window of MATLAB. However, users should try to avoid interactively working in the GUI mode of any scientific program at the HPC cluster. This is because the HPC cluster is used for computation purposes only and not for developing or debugging work. Users should always submit their jobs to the queuing system in the batch mode. If you really need to work interactively, please submit an interactive jobs to the queue system (See Sec. 4.2.4).

Create job script file for the queue system

Now you can create the job script file by using the text editors as mentioned above. For example, we could run '`nano run.sh`' to open a new file named '`run.sh`' and type in the following contents:

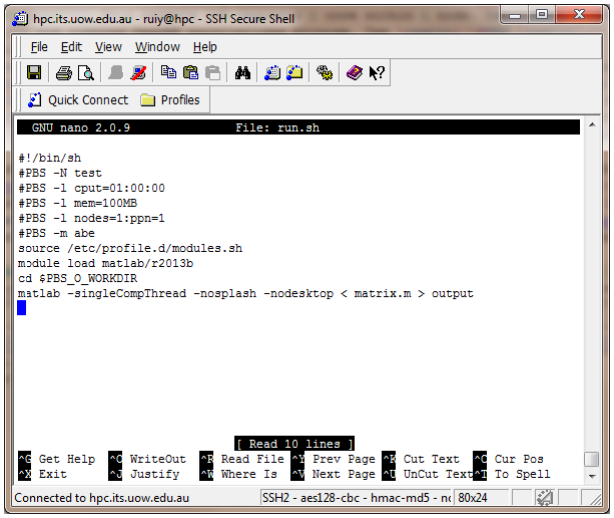
hpc.its.uow.edu.au - ruiy@hpc - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
GNU nano 2.0.9 File: run.sh
#!/bin/sh
#PBS -N test
#PBS -l cput=01:00:00
#PBS -l mem=100MB
#PBS -l nodes=1:ppn=1
#PBS -m abe
source /etc/profile.d/modules.sh
module load matlab/r2013b
cd $PBS_O_WORKDIR
matlab -singleCompThread -nosplash -nodesktop < matrix.m > output
[Read 10 lines]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
Connected to hpc.its.uow.edu.au
SSH2 - aes128-cbc - hmac-md5 - n
80x24

Figure 2.4.: Use `nano` to create your job script file.

The information contains several job control flags starting with '`#PBS`' and several lines to execute the program. Each line is explained as below:

- `#!/bin/sh`
  The shell environment in use, don't change.
- `#PBS -N test`
  Use '`-N`' flag to specify job name, change 'test' to your preferred job name.
- `#PBS -m abe`
  Specify the email notification when the job aborts(a), begins(b) and/or finishes(e). Delete this line if you don't want to receive email notification.
- `#PBS -l cput=01:00:00`
  Use '`-l`' flag to request resources such as executing time, memory, cpu numbers etc. Here, requests executing time as 1 cpu hour in the 'hh:mm:ss' format. The

maximum cpu time you can request is 1000 hours, i.e. '1000:00:00'.
NOTE: The default cpu time is 48 hours if no 'cput' is defined.

- `#PBS -l mem=100MB`
  Request 100MB memory here. Other units of '`mem`' could be 'GB'.
- `#PBS -l nodes=1:ppn=1`
  This example job requests 1 core within 1 node. Use '`#PBS -l nodes=1:ppn=`*N*' to request *N* ($\leq$32) cores within a single node if you are running parallelized program.
- `source /etc/profile.d/modules.sh`
  Invoke the module environment, always keep this line before loading any other application module.
- `module load matlab/r2013b`
  Load the module of 'MATLAB' application in version R2013b. This will set the appropriate environment to run the program.
- `cd $PBS_O_WORKDIR`
  This changes the current working directory to the directory from which the script was submitted.
- `matlab -singleCompThread -nosplash -nodesktop < matrix.m > output`
  This line will execute the program in the command line mode. The flag varies along with different application in use. Use '<' to get the matlab read the input file 'matrix.m' and use '>' to pipe the results into the file '`output`'. Please always use '`-singleCompThread -nosplash -nodesktop`' flag to run a MATLAB batch job.

Please note, the job script should not execute directly at the command line, but must be submitted by following the '`qsub`' command. Please refer to the Sec. 4.2.1 for detailed usage on the PBS control flags. Applications installed at the HPC cluster can be loaded by using the '`module load`' command as shown in the above script file. To check those avaiable software packages, type '`module avail`' in the command line. Please refer to Appendix F for detailed information on how to use each software packages installed on the cluster. You can find job script examples for a variety of software under the directory '`/hpc/tmp/examples`'.

2.2.4. Submit your job and check the results

Now you are ready to submit your job script to the queue system. Please follow the procedure as below (Each step corresponds to a '`-sh-4.1$`' prompt line in Fig. 2.5.)

1. First make sure you are working under the '`/USERHOME/task`' directory by typing '`pwd`'. You can type "`cd`' to return to your home directory wherever your are.
2. Next make sure you have the two initialization files 'matrix.m' and 'run.sh' within this directory by typing '`ls`'.
3. Now submit the job script 'run.sh' to the queue system by typing '`qsub run.sh`' and you will be returned a job id (5785 in this example) by the queue system.
4. Check whether the job has been completed by using the '`qstat`' command, i.e.

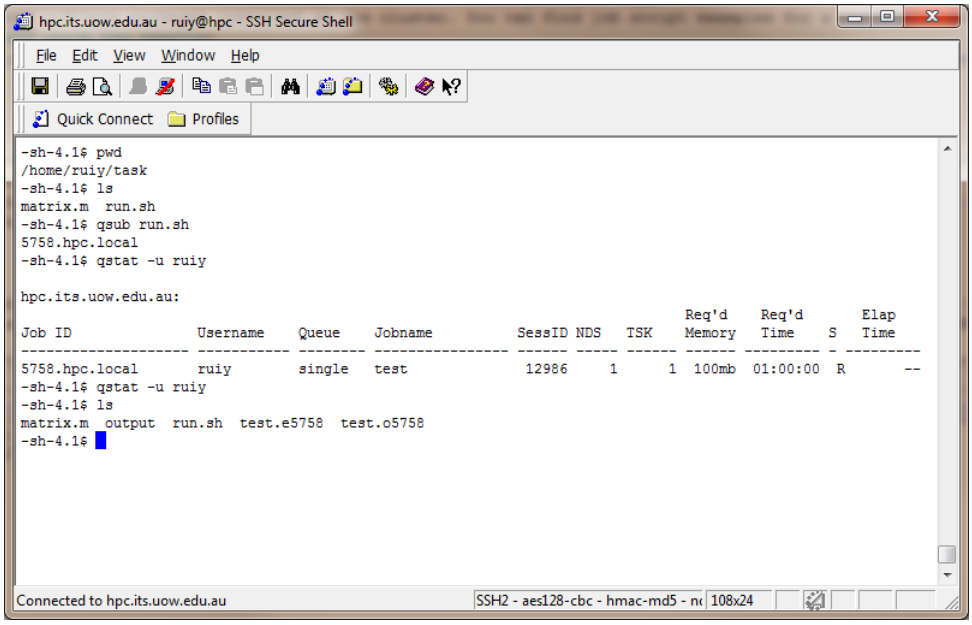
hpc.its.uow.edu.au - ruiy@hpc - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
-sh-4.1$ pwd
/home/ruiy/task
-sh-4.1$ ls
matrix.m run.sh
-sh-4.1$ qsub run.sh
5758.hpc.local
-sh-4.1$ qstat -u ruiy
hpc.its.uow.edu.au:
Job ID Username Queue Jobname SessID NDS TSK Req'd Memory Req'd Time S Elap Time
5758.hpc.local ruiy single test 12986 1 1 100mb 01:00:00 R --
-sh-4.1$ qstat -u ruiy
-sh-4.1$ ls
matrix.m output run.sh test.e5758 test.o5758
-sh-4.1$
Connected to hpc.its.uow.edu.au
SSH2 - aes128-cbc - hmac-md5 - n
108x24

Figure 2.5.: Submit your job.

```
qstat -u YOUR_USERNAME
```

Alternatively you could just type '`qstat -a`' as users are only allowed to view their own jobs. If there is something printed on the screen as shown in Fig. 2.5, the job is either queuing(Q) or running(R). This command tells you the job id, the user who submitted it, the name of queue running it, job name, session ID, how many nodes in use, how many cores are in use, the requested memory, the requested CPU time, the status of the job and the elapsed time so far.
NOTE If you realise there is something wrong while the job is running, you could use the command '`qdel` *JobID*' to delete your own job, i.e. '`qdel 5758`' and fix the problem.

5. Check the job status from time to time by repeating the command '`qstat -a JOBID`'. If there is no messages printed any more, your job completes. You will also receive a notification email about the job completion if '`#PBS -m abe`' was set in the job script.
6. Three new files will be generated when you run the job, 'output' is created by the program MATLAB which contains the calculation results, 'test.e5758' and 'test.o5758' are created by the queue system which contains the standard error and output message respectively.
7. You could chose a text editor such as 'nano', 'gedit', 'vi' or 'emacs' etc. to check the contents of these 3 files. An alternative and prompt way to view a '`text`' file is to use a Linux intrinsic command '`more`'. Type '`more output`' in the command line to check the content of the file 'output'. As shown in the Fig. 2.6, the matrix multiplication between A and B was printed in addition to some 'MATLAB' program messages.

8. Although the job finished normally, the warning message may printed in the standard error message file i.e. 'test.e5758' created by the queue system. Also because we

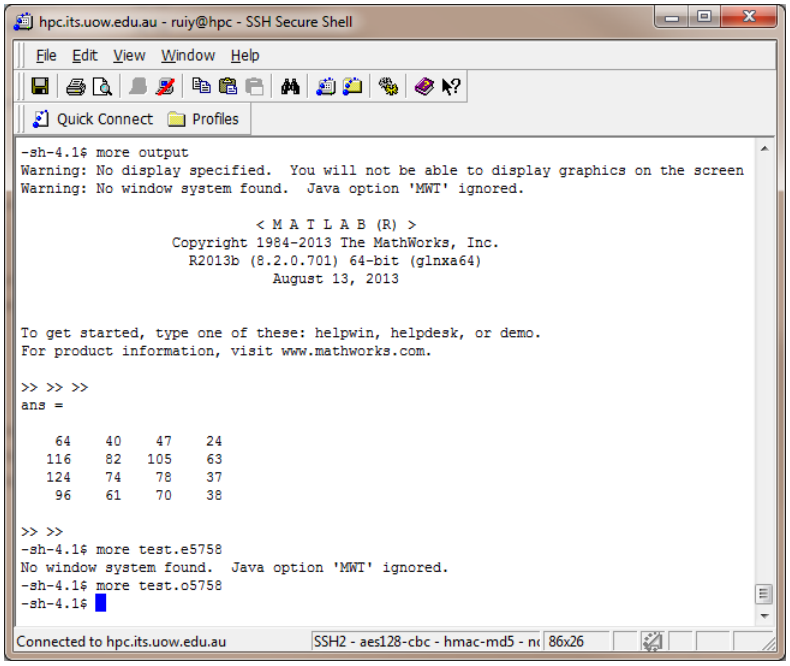
hpc.its.uow.edu.au - ruiy@hpc - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
-sh-4.1$ more output
Warning: No display specified. You will not be able to display graphics on the screen
Warning: No window system found. Java option 'MWT' ignored.
< M A T L A B (R) >
Copyright 1984-2013 The MathWorks, Inc.
R2013b (8.2.0.701) 64-bit (glnxa64)
August 13, 2013
To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.
>> >> >>
ans =
64 40 47 24
116 82 105 63
124 74 78 37
96 61 70 38
>> >>
-sh-4.1$ more test.e5758
No window system found. Java option 'MWT' ignored.
-sh-4.1$ more test.o5758
-sh-4.1$
Connected to hpc.its.uow.edu.au
SSH2 - aes128-cbc - hmac-md5 - n
86x26

Figure 2.6.: Check the results.

redirect the result to the 'output' file by using '>', there are no results sent to the standard I/O message file, i.e. 'test.o5758'.

At this point, you have successfully completed a computational task at the HPC cluster. Please note, the information introduced in this document is limited just to run a simple task. Please refer to the Chapter 4 for general job submissions and check Appendix F to prepare the job script of a specific package. Please also read other chapters of this manual for further information on how to work smoothly at the HPC cluster.

3. Software

3.1. Software Installation

There are many software packages have been deployed at the HPC cluster and will be kept upgrading by the system administrator. If user requires a software that is not installed at the HPC cluster please send your request to the HPC admin. Users can also install the software under their own home directory which is solely accessed by the user.

Note As there is no central budget for software licenses, any costs incurred for the requested software must be covered by the department of the university that the user is employed by or the user's research groups. Please check Sec. 3.3 for the available packages at the HPC cluster.

3.2. Software Environment

The package `Environment Modules` is deployed at the HPC cluster to allow easy customization of user's shell environment to the requirements of whatever software you wish to use. The `module` command syntax is the same no matter which command shell you are using as listed in the following:

- `module avail` will show you a list of the software environments which can be loaded via '`module load package`' command. Example:

```
-sh-4.1$ module avail

--------------------------- /usr/share/Modules/modulefiles ------------------------
dot               module-info      null             rocks-openmpi_ib
module-cvs        modules          rocks-openmpi    use.own

---------------------------------- /etc/modulefiles ------------------------------
R/3.0.2                            mpich/3.0.4_gcc
ambertools/13_gcc                  mpich/3.0.4_itl
atsas/2.5.1-1                      mpich/3.0.4_open64
autodock_vina/1.1.2_bin            mpich/3.0.4_pgi
autodocksuite/4.2.5.1              mrbayes/3.2.2
byacc/20130925                     namd/2.9_bin(default)
cdo/1.6.1                          ncl/6.1.2
cmake/2.8.12                       nco/4.3.7
cp2k/2.4_gcc                       nose/1.3.0
cp2k/2.4_gcc_acml                  numpy/1.8.0
cpmd/3.17.1_pgi                    open64/5.0
dock/6.5                           openmpi/1.4.4_gcc
dock/6.5_mpi                       openmpi/1.4.4_pgi
eclipse_pad/kepler_sr1             openmpi/1.6.5_dbg
fftw/2.1.5_itl                     openmpi/1.6.5_gcc
fftw/3.3.3_gcc                     openmpi/1.6.5_itl
fftw/3.3.3_gcc_4.8.2               openmpi/1.6.5_open64
fftw/3.3.3_itl                     openmpi/1.6.5_pgi
fftw/3.3.3_pgi                     orca/2.9.1
gamess/May_2013_R1                 orca/2.9.1_pgi
gaussian/g09a02(default)           orca/3.0.0
gaussian/g09c01                    orca/3.0.0_dbg
```

```
gcc/4.7.1                           orca/3.0.0_pgi
gcc/4.8.2                           orca/3.0.1_pgi
geant/4.9.5p02                      pdb2pqr/1.8
geant/4.9.6p01                      pgi/13.9
geant/4.9.6p02                      propka/3.1
gnuplot/4.6.4                       pyqt/4.10.3
idl/8.2_sp3                         qchem/4.1.0
intel_ics/2013_sp1                  qiime/1.7.0
intel_mkl/11.1                      quantum_espresso/5.0.3_openmpi_pgi
intel_mpi/2013_sp1                  repasthpc/2.0
ipython/1.1.0                       root/5.34.05
matlab/r2011a                       rosetta/3.5
matlab/r2013a                       scipy/0.13.0
matlab/r2013b                       sumo/0.18.0
matplotlib/1.3.1                    vmd/1.9.1
mgltools/1.5.6                      wien2k/12.1_itl_mkl
mpich/3.0.4_dbg
```

- `module load` *`package`* will load the software environments for you. Example:

```
-sh-4.1$ module load R/3.0.2
```

- `module help` *`package`* should give you a little information about what the '`module load` *`package`*' will achieve for you. Example:

```
-sh-4.1$  module help R/3.0.2

----------- Module Specific Help for 'R/3.0.2' --------------------

This modulefile provides R (3.0.2, x86-64)

More information about R can be found at:
    http://www.r-project.org/

-sh-4.1$
```

- `module show` *`package`* will detail the command in the module file. Example:

```
-sh-4.1$  module show  R/3.0.2
-------------------------------------------------------------------
/etc/modulefiles/R/3.0.2:

module-whatis    Sets the environment for R (3.0.2, x86-64)
conflict         R
append-path      PATH /hpc/software/package/R/3.0.2/bin
append-path      MANPATH /hpc/software/package/R/3.0.2/share/man
-------------------------------------------------------------------

-sh-4.1$
```

- `module list` prints out those loaded modules. Example:

```
-sh-4.1$ module list
Currently Loaded Modulefiles:
 1) R/3.0.2
```

- `module unload` *`package`* will unload those loaded modules. Example:

```
-sh-4.1$ module unload R/3.0.2
-sh-4.1$ module list
No Modulefiles Currently Loaded.
-sh-4.1$
```

NOTE: The available software packages are subject to change and keep upgrading from time to time. Please check the latest version of all available packages by using '`module avail`' command.

3.3. Software List

Some packages installed at the UOW HPC cluster are licensed and paid for by various departments and research groups on campus. The '*' under the 'Prerequisite' column indicates software that requires some discussion with the HPC admin to be able to access.

NOTE: All installed software packages have passed the standard test suite (if existing). However, it is user's responsibility to check the correctness and the validity of the software deployed at the cluster prior to publishing any result by using them.

NOTE: Please add '`source /etc/profile.d/modules.sh`' before any module command in your job script when submit it to the queue system.

Table 3.1.: List of Software Packages

Name	Version	Prerequisite
	Parallel Programming Libraries/Tools (F.1)	
Intel MPI	2013_sp1 2015 2015_sp2	
MPICH	1.5 3.0.4	
OpenMPI	1.4.4 1.6.5	
	Compilers & Building Tools (F.2)	
CMake	2.8.12	
GCC	4.4.7 4.7.1 4.8.2 4.9.2	
Intel Compiler	2013_sp1 2015 2015_sp2	
Open64	5.0	
PGI Compiler	13.9 14.7 15.3	
	Scripting Languages (F.3)	
iPython	1.1.0	
Java	1.7.0_13	
Perl	5.10.1 5.20.2	
Python	2.6.6 2.7.6	
	Code Development Utilities (F.4)	
Eclipse for Parallel Application Developers	kepler_SR1 kepler_SR2	
	Math Libraries (F.5)	
ACML	5.3.1 6.1.0	
ATLAS	3.11.17	
BLACS	—	
BLAS,CBLAS	3.5.0	
Boost	1.55.0	
FFTW	2.1.5 3.3.3 3.3.4	
GMP	4.3.2 5.1.3	

Continued on next page

Table 3.1 – continued from previous page

Name	Version	Prerequisite
GNU Scientific Library	1.16	
Intel MKL	11.1 11.2	
LAPACK	3.4.2 3.5.0	
MPFR	2.4.2 3.1.2	
NumPy	1.8.0 1.8.1 1.8.1_py27 1.9.1	
SCALAPACK	2.0.2	
Scipy	0.13.0 0.14.0_py27 0.15.1	
Debuggers,Profilers and Simulators (F.6)		
HPCToolkit	3.5.2	
TAU	2.24	
Valgrind	3.9.0 3.10.0	
Visualization (F.7)		
GNUPlot	4.6.4	
IDL	8.2_sp3 8.3	*
Matplotlib	1.3.1 1.4.2	
Ncar Command Language	6.1.2	
Ncview	2.1.4	
OpenCV	2.4.8	
Statistics and Mathematics Environments (F.8)		
R	3.0.2 3.1.1	
Computational Physics&Chemistry (F.9)		
ABINIT	7.8.2 7.10.2	
ASE	3.8.1	
Atomistix ToolKit(ATK)	13.8.1	*
AutoDock	4.2.5.1	
AutoDock Vina	1.1.2	
CP2K	2.4 2.5.1 2.6.0	
CPMD	3.17.1	*
DOCK	6.5	*
GAMESS	May_2013_R1	*
GATE	6.2 7.0	
GAUSSIAN	g09a02 g09c01	*
Geant	4.9.5p02 4.9.6p01 4.9.6p02 4.9.6p03 4.9.6p04 4.10.0 4.10.0p01 4.10.0p02 4.10.0p03 4.10.1	
GPAW	0.10.0	

Continued on next page

Table 3.1 – continued from previous page

Name	Version	Prerequisite
GROMACS	4.6.5	
MGLTools	1.5.6	
Molden	5.1.0	
NAMD	2.9 2.10	*
NWChem	6.3 R2	
OpenBabel	2.3.2	
ORCA	3.0.1 3.0.2 3.0.3	
Q-Chem	4.0.0.1 4.2.0	*
Quantum_Espresso	5.0.3 5.1.0 5.1.1 5.1.2	
Siesta	3.2 p4 3.2 p5	
VMD	1.9.1 1.9.2	*
Wien2K	12.1	*
XCrySDen	1.5.53	
	Informatics (F.10)	
Caffe	20140616 20150216	
netCDF C library	4.3.0	
netCDF C++ Library	4.2.1	
netCDF Fortran Library	4.2	
netCDF Operator	4.3.7	
RepastHPC	2.0	
SUMO	0.18.0	
	Engineering (F.11)	
ANSYS	14.5	*
ANSYS Electromagnetics Suite	16.0	*
ABAQUS	6.9-1 6.12-1	*
LAMMPS	5Nov10 1Dec13 30Oct14	
Materials Studio	7.0	*
MATLAB	R2011a R2013a R2013b R2014a R2014b	
	Biology (F.12)	
ATSAS	2.5.1-1	
MrBayes	3.2.2	
PartitionFinder	1.1.1	
QIIME	1.7.0 1.8.0	

Check Appendix F to learn how to use packages at the HPC cluster in detail.

4. Queue System

4.1. Queue Structure

The computing resources of the HPC cluster are under the management of the Torque resource manager (also known as PBS) and are scheduled by the Maui scheduler. All computational jobs should be executed via the queue system. Users submit jobs to a queue by specifying the number of CPUs, the amount of memory, and the length of time needed (and, possibly, other resources). The 'Maui' scheduler then run the job according to its priority when the resources are available, subject to constraints on maximum resource usage. 'Maui' is capable of very sophisticated scheduling and will be tuned over time to meet the requirements of the user community while maximizing overall throughput.

4.1.1. Normal queue

The default queue is a routine queue called 'normal' which the queue system uses to look at your job and figure out into which execution queue it should actually go. At the time of this writing, your job's cpu number request determines this. Users do not need to specify the execution queue as the system will figure it out. There are 5 execution queues can be arrived by the default 'normal' queue:

Table 4.1.: Queue structure

Queue	**CPU Core Limit**	**CPU Time Default**	**CPU Time Limit**	**Wall Time Limit**
single	1	48:00:00	1000:00:00	1000:00:00
para_4	2∼4	48:00:00	1000:00:00	256:00:00
para_8	8	48:00:00	1000:00:00	128:00:00
para_16	16	48:00:00	1000:00:00	64:00:00
para_32	32	48:00:00	1000:00:00	32:00:00

All above execution queues allow jobs running up to 1000 hours cpu time (i.e. the accumulated cpu running time) or the corresponding wall time limit. As each compute node contains 64 cores, please always request CPU resources within a single node.
NOTE: The default cpu time is 48 hours for all above normal queues. Please specify 'cput' in your job script to request the longer cpu time.

4.1.2. Special queues

In addition to the default 'normal' queue, there are 2 more execution queues for running jobs with special requests:

- **short** allows interactive or batch jobs requesting 1 core to run up to 48 hour;
- **long** allows batch jobs running with the wall time over 2000 hours;

The above two queues are not included in the 'normal' queue. Users should manually specify either of the above queues in their job script to submit a job to them. The 'short' is the only queue enabling users to run single-core interactive jobs at the cluster with up to 48 hours. Please remember to type 'exit' to quit the interactively job upon finishing the work. The 'long' queue is used for users running long-term jobs which may exceed 1000 CPU hours. This is useful for running applications which can not restart the work from the checkpoint files. Please note each users can only utilize a very limited CPU cores in the 'long' queue and the 'long' queue has the lower priority to get resources than other execution queues.

4.1.3. Schedule policy

The basic scheduling policy is FIFO (first in first out) within each queue, i.e. queuing jobs in the order that they arrive. However, the less resources (walltime, cput, mem, nodes etc.) requested in the job, the higher priority that the job to be put into run. The job priority also depends on several fairness share factors, such as the user's recent utilization, job queuing time and the queue priority etc. Since the job is allocated based on the resources requested and available, please make reasonable requests to help your job beginning sooner.

4.2. Job Management

4.2.1. PBS options

To run a job at the HPC cluster, users must write their own job script and submit it to the queue system. The job script is an ascii file containing the PBS directives and the shell script to run commands and programs. The first primary task of the job script is to request the computing resource via PBS directives at its head. The PBS directives are all starting with '`#PBS`' and are all at the beginning of the script, that there are no blank lines between them, and there are no other non-PBS commands until after all PBS directives.

```
#!/bin/sh
#PBS -N test_job
#PBS -l nodes=1:ppn=1
#PBS -l cput=03:00:10
#PBS -l mem=400MB
#PBS -q normal
#PBS -o test.out
#PBS -e test.err
#PBS -m abe
#PBS -V
```

Explanations on the above PBS directives are listed below:

- `#!/bin/sh`
  The shell environment in use.

- `#PBS -N test_job`
  Use `-N` flag to specify name of the job.
- `#PBS -l nodes=1:ppn=1`
  Use `-l` flag to request resources. This example job requests 1 core (ppn) within a single node (nodes). Use '`#PBS -l nodes=1:ppn=N`' to request N ($\leq$32) cores within the same node if you are running parallelized program.
- `#PBS -l cput=03:00:10`
  Request 3 hours cpu time in the format of `hh:mm:ss`. Users can also request Walltime time with `walltime`.
- `#PBS -l mem=400MB`
  Request 400MB memory for the job. Users can also request memory in the following formats
  - `vmem` total virtual memory for the job;
  - `pvmem` virtual memory per core;
  - `pmem` memory per core;

  The units of the memory requests could be `MB` and `GB`.
  NOTE: Make sure to request enough memory in the job. If the job uses more memory than the request, it will be firstly suspended and then deleted. If you are not clear how much memory the job will consume, just remove any memory request in the job script- the system will assign the maximum allowed memory for the job. The maximum allowed memory is normally 4GB per core but may vary on special queues.
- `#PBS -q normal`
  Use `-q` flag to specify the destination queue of the job. This line can be omitted when the job is submitted to the 'normal' queue. You have to specify the queue name when submit jobs to either 'short' or 'long' queues, i.e.

  ```
  #PBS -q short
  ```

  or

  ```
  #PBS -q long
  ```

- `#PBS -o test.out`
  Use `-o` flag to specify the name and path of the standard output file.
- `#PBS -e test.err`
  Use `-e` flag to specify the name and path of the standard error file.
- `#PBS -m abe`
  Use `-m` flag to specify the email notification when the job aborts, begins and/or finishes.
- `#PBS -V`
  Export user's environment variables to the job.

Other PBS flags are listed as below:

- `#PBS -j [eo|oe]` Merge `STDOUT` and `STDERR`. If '`eo`' merge as standard error; if '`oe`' merge as standard output.
- `#PBS -v` Customize the user defined variables.

Type '`man qsub`' for more details on using '`qsub`' command.

When a batch job starts execution, a number of environment variables are predefined, which include:

- variables defined on the execution host.
- variables exported from the submission host with '`-v`' (selected variables) and '`-V`' (all variables).
- variables defined by PBS.

The following variables reflect the environment where the user ran `qsub`:

- `PBS_O_HOST` The host where you ran the qsub command.
- `PBS_O_LOGNAME` Your user ID where you ran qsub.
- `PBS_O_HOME` Your home directory where you ran qsub.
- `PBS_O_WORKDIR` The working directory where you ran qsub.

These variables reflect the environment where the job is executing:

- `PBS_ENVIRONMENT` Set to `PBS_BATCH` to indicate the job is a batch job, or to `PBS_INTERACTIVE` to indicate the job is a PBS interactive job.
- `PBS_O_QUEUE` The original queue you submitted to.
- `PBS_QUEUE` The queue the job is executing from.
- `PBS_JOBID` The job's PBS identifier.
- `PBS_JOBNAME` The job's name.

4.2.2. Submit a batch job

Another primary task of the job script is to run the program properly which can be accomplished by a batch of shell scripts. Please note user's home directory and the extended space are mounted at the remote network file system (NFS). Running jobs directly under these directories may produce heavy traffic over the network and make the storage server overloaded. To reduce the load at the storage server and enhance the job performance, users should always run their jobs by using the local scratch disks located at each compute node especially when the job frequently reads/writes a large volume of scratch files.

A job script template of using the local scratch disks is shown as below:

```
#!/bin/sh
#PBS -N job
#PBS -l cput=01:00:00
#PBS -l pvmem=4GB
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe
#PBS -l nodes=1:ppn=1

#=====================================================#
# USER CONFIG
#=====================================================#
INPUT_FILE="****"
OUTPUT_FILE="****"
MODULE_NAME="****"
PROGRAM_NAME="****"
# Set as true if you need those scratch files.
COPY_SCRATCH_BACK=true

#=====================================================#
```

```
21 # MODULE is loaded
22 #=========================================================#
23 NP=`wc -l < $PBS_NODEFILE`
24 source /etc/profile.d/modules.sh
25 module load $MODULE_NAME
26 cat $PBS_NODEFILE
27
28 #=========================================================#
29 # SCRATCH directory is created at the local disks
30 #=========================================================#
31 SCRDIR=/export/scratch/common/$PBS_JOBID
32 if [ ! -d "$SCRDIR" ]; then
33 mkdir $SCRDIR
34 fi
35
36 #=========================================================#
37 # TRANSFER input files to the scratch directory
38 #=========================================================#
39 # just copy input file
40 cp $PBS_O_WORKDIR/$INPUT_FILE $SCRDIR
41 # copy everything (Option)
42 #cp $PBS_O_WORKDIR/* $SCRDIR
43
44 #=========================================================#
45 # PROGRAM is executed with the output or log file
46 # direct to the working directory
47 #=========================================================#
48 echo "START TO RUN WORK"
49 cd $SCRDIR
50
51 # Run a system wide sequential program
52 $PROGRAM_NAME < $INPUT_FILE >& $PBS_O_WORKDIR/$OUTPUT_FILE
53 # Run a MPI program (Option)
54 # mpirun -np $NP $PROGRAM_NAME < $INPUT_FILE >& $OUTPUT_FILE
55 # Run a OpenMP program(Option)
56 # export OMP_NUM_THREADS=$NP
57 # $PROGRAM_NAME < $INPUT_FILE >& $OUTPUT_FILE
58
59 #=========================================================#
60 # RESULTS are migrated back to the working directory
61 #=========================================================#
62 if [[ "$COPY_SCRATCH_BACK" == *true* ]]
63 then
64     echo "COPYING SCRACH FILES TO " $PBS_O_WORKDIR/$PBS_JOBID
65     cp -rp $SCRDIR/* $PBS_O_WORKDIR
66     if [ $? != 0 ]; then
67         {
68             echo "Sync ERROR: problem copying files from $tdir to $PBS_O_WORKDIR;"
69             echo "Contact HPC admin for a solution."
70             exit 1
71         }
72     fi
73 fi
74
75 #=========================================================#
76 # DELETING the local scratch directory
77 #=========================================================#
78 cd $PBS_O_WORKDIR
79 if [[ "$SCRDIR" == *scratch* ]]
80 then
81     echo "DELETING SCRATCH DIRECTORY" $SCRDIR
82     rm -rf $SCRDIR
83     echo "ALL DONE!"
84 fi
85 #=========================================================#
86 # ALL DONE
87 #=========================================================#
```

There are 8 portions in the above job script:

- Lines 1∼8: Request the computing resources (see detail in Sec. 4.2.1).
- Lines 10∼18: '**USER CONFIG**' portion enables users to specify the job running

parameters. Users should replace the '****' characters with appropriate values, such as the input file name for `INPUT_FILE`, output file name for `OUTPUT_FILE`, the module name to be loaded for `MODULE_NAME` and the actual program name for `PROGRAM_NAME`. There are two kinds of programs:

  - The system wide program as listed in the '`module avail`' command: specify the module name and the program name&flags to both `MODULE_NAME` and `PROGRAM_NAME` respectively.
  - The customized program built by the user: give the full path of the program name&flags to `PROGRAM_NAME`.

  As default, the system will copy all scratch files back to the working directory after the job termination. Set `COPY_SCRATCH_BACK` as 'false' if you do not want the scratch files being copied back.

- Lines 20∼26: '**`MODULE`**' portion loads the necessary module for the job.
- Lines 28∼34: '**`SCRATCH`**' portion setup the scratch directory at the local disks.
- Lines 36∼42: '**`TRANSFER`**' portion transfers input files to the scratch directory. Users can also copy all files under the working directory to the local scratch directory by removing '#' at the beginning of the Line 42.
- Lines 44∼54: '**`PROGRAM`**' portion runs the program according to the users' settings at the '**`USER CONFIG`**'. There are several scenarios to run the program:
  - The sequential program using a single core: use the line 52 and comment line 53-57 (put "#" at the beginning of each line).
  - The parallelized program using MPI: enable the line 54 by removing its beginning "#" and comment line 51-53 and 55-57.
  - The parallelized program using OpenMP: enable line 56 and 57 by removing their "#" characters and comment line 51-55.

  Make sure there is only one of the line 52, 54 and 56 is enabled.
- Lines 56∼63: '**`RESULTS`**' portion copy scratch files back to the user's working directory if the '`COPY_SCRATCH_BACK`' at the '**`USER CONFIG`**' is set as `true`.
- Lines 65∼74: '**`DELETING`**' portion delete the local scratch directory.

Usually the user just revises the PBS directives and the contents of '**`USER CONFIG`**' without touching other portions. However, in some special circumstances users need to revise the whole script to match the job request.
After producing the job script, run the `qsub` command to submit it to the queue system, i.e.

```
-sh-3.2$ qsub jobscript
```

A job identifier (Job ID) will be given after the job being suuccessfuly submitted.

4.2.3. Check the job/queue status

Job progress can be monitored using command '`qstat -a`'. It will give the following information: job identifier, job name, username, elapsed CPU time, job status and the queue in which the job resides. Status can be one of the following:

- `E` - job is exiting after having run
- `H` - job is held

- `Q` - job is queued, eligible to be run or routed
- `R` - job is running
- `T` - job is being moved to new location
- `W` - job is waiting for its execution time to be reached
- `S` - job is suspended

```
-sh-4.1$ qstat -a

hpc.its.uow.edu.au:
                                                                  Req'd      Req'd Elap
Job ID          Username  Queue    Jobname  SessID NDS  TSK  Memory Time     S Time
--------------- --------- -------- -------- ------ ---- ---- ------ -------- - ----------
1094.hpc.local  ruiy      para_8   test_job 8703   1     8   --     60:00:00 R 00:00:15
```

Other `qstat` flags:

- `qstat -u` *`username`* Display all jobs belong to a specific user.
  For example '`qstat -u ruiy`' will check the status of jobs belong to user '`ruiy`'. This flag can be omitted at the HPC cluster as normal users can only view their own jobs.
- `qstat -f` *`jobid`* Full display of a job with a specific jobid.
  For jobs running in the parallel mode, users could check the job efficiency by comparing the values of

  ```
  resources_used.cput
  ```

  and

  ```
  resources_used.walltime
  ```

  from the output. Generally, the value of the `resources_used.cput` should be around

  ```
  50∼99%×(resources_used.walltime × requested_CPU)
  ```

  Otherwise, there might be unmatched CPU cores between the input file and the job script. In this case, check both the input file and the job script file to make the requested CPU cores consistent with those in use.
- `qstat -Q` Display the queue status.

  ```
  -sh-4.1$ qstat -Q
  Queue            Max    Tot   Ena   Str   Que   Run   Hld   Wat   Trn   Ext T   Cpt
  ---------------- ---    ----   --    --   ---   ---   ---   ---   ---   --- -   ---
  batch              0      0    no   yes     0     0     0     0     0     0 E     0
  short             16      0   yes   yes     0     0     0     0     0     0 E     0
  para_16           24     26   yes   yes     1    14    11     0     0     0 E     0
  normal           512      0   yes   yes     0     0     0     0     0     0 R     0
  para_64            0      0    no   yes     0     0     0     0     0     0 E     0
  long              48     38   yes   yes    14    24     0     0     0     0 E     0
  para_m            10      0   yes   yes     0     0     0     0     0     0 E     0
  single           360    291   yes   yes    45   246     0     0     0     0 E     0
  para_8            48    155   yes   yes     0     8   147     0     0     0 E     0
  para_32           12      0   yes   yes     0     0     0     0     0     0 E     0
  para_4            48      0   yes   yes     0     0     0     0     0     0 E     0
  ```

  The following information of all available queues are displayed:
  - `Queue` the queue name
  - `Max` the maximum amount of nodes that a job in the queue may request
  - `Tot` number of jobs currently in the queue

- `Ena` queue is enabled (yes) or disabled (no)
- `Que` number of 'queued 'jobs
- `Run` number of 'running' jobs
- `Hld` number of 'held' jobs

Users could request appropriate resources to reduce the job queueing time based on the above queue utilization information. Please note, some queues are disabled as they are only for testing purpose.

Type '`man qstat`' for more details on using '`qstat`' command.

4.2.4. Submit an interactive job

Interactive batch jobs are likely to be used for debugging large or parallel programs and especially for running time-consuming, memory consuming and I/O consuming commands. It uses the CPU and memory of a computer node which can largely reduce the work load on the login node. An example on working with the interactive job is shown below:

Suppose user `ruiy` is working at the login node, i.e.

```
-sh-4.1$ hostname
hpc.its.uow.edu.au
```

Submit an interactive job to the 'short' queue

```
-sh-4.1\$ qsub -I -q short
qsub: waiting for job 54718.hpc.local to start
qsub: job 54718.hpc.local ready

-sh-4.1$
```

An interactive shell being started out at one of the compute node once the job starts

```
-sh-4.1$ hostname
hpcn01.local
```

and initially login the user's home directory

```
-sh-4.1$ pwd
/home/ruiy
```

Check the job status

```
-sh-4.1$ qstat

hpc.its.uow.edu.au:
                                                                  Req'd   Req'd Elap
Job ID               Username Queue  Jobname  SessID NDS  TSK Memory Time  S Time
-------------------- -------- ------ -------- ------ ---- --- ------ ----- - ----------
54718.hpc.local      ruiy     short  STDIN    11701  1    1   429496 48:00 R --
```

Terminate the job

```
-sh-3.2$ exit
logout

qsub: job 54718.hpc.local completed
```

Return to the login node

```
-sh-4.1$ hostname
hpc.its.uow.edu.au
```

A submission script cannot be used in this mode - the user must provide all `qsub` options on the command line for the interactive job. The submitted interactive jobs are subject to all the same constraints and management as any other job in the same queue. Don't forget to complete the interactive batch session by typing '`exit`' to avoid leaving cpus idle on the machine.

To submit an interactive job which enables the GUI package, add the '`-X`' flag as below

```
-sh-4.1$ qsub -I -X -q short
```

Please note, the user also needs to login in the cluster by specifying '-X' flag in '`ssh`' command when using Linux desktop or enable X-server at the Windows desktop.
Note Users can only submit their interactive jobs to the 'short' queue.

4.2.5. Submit workflow jobs

In some cases a single simulation requires multiple long runs which must be processed in sequence. For this purpose, uses can use the '`qsub -W depend=...`' options to create dependencies between jobs.

```
qsub -W depend=afterok:<Job-ID> <QSUB SCRIPT>
```

Here, the batch script `<QSUB SCRIPT>` will be submitted after the Job `<Job-ID>` was successfully completed. Useful options to '`depend=...`' are

- `afterok:<Job-ID>` Job is scheduled if the Job `<Job-ID>` exits without errors or is successfully completed.
- `afternotok:<Job-ID>` Job is scheduled if the Job `<Job-ID>` exited with errors.
- `afterany:<Job-ID>` Job is scheduled if the Job `<Job-ID>` exits with or without errors.

By using the command, we can tell the queue system how our jobs depend on other jobs, so that the queue system will wait for the first job to finish before releasing the second job. Then the queue system will wait for the second job to finish, before the third job gets released, and so on.

```
-sh-4.1$ qsub run1.sh
5977.hpc.local
-sh-4.1$ qsub -W depend=afterok:5977.hpc.local run2.sh
5978.hpc.local
-sh-4.1$ qsub -W depend=afterok:5978.hpc.local run3.sh
5979.hpc.local
-sh-4.1$ qstat -a
```

```
hpc.its.uow.edu.au:
                                                            Req'd  Req'd       Elap
Job ID          Username Queue    Jobname SessID NDS  TSK   Memory Time      S Time
--------------- -------- -------- ------- ------ ---- ---- ------ --------  - --------
5977.hpc.local ruiy     single  run1.sh 61829  1    1    --     48:00:00 R 00:00:00
5978.hpc.local ruiy     single  run2.sh --     1    1    --     48:00:00 H      --
5979.hpc.local ruiy     single  run3.sh --     1    1    --     48:00:00 H      --
```

Viewing these jobs in the queue will show the first submitted job state (S column) as 'R' for running. The succeeding ones will have a job state of 'H' for held, because they are dependent on the first job.

4.2.6. Delete jobs

To delete a specific batch job, type '`qdel` *`jobid`*' in the command line where *jobid* is the job's identifier, produced by the `qsub` command. However, the command has no effect on an interactive job and the user needs to type '`exit`' to quit it.

Type '`man qdel`' for more details on using '`qdel`' command.

To delete all jobs belong to a user, use the following shell commands:

```
qselect -u $USER | xargs qdel
```

`qselect` prints out a job list based on specific criterions, `xargs` takes multiple line input and run the command you give to it repeatedly until it has consumed the input list. `$USER` is your own user name and you can safely use your own user name to replace `$USER` in the above command. A user can not delete jobs belong to other users.

Delete all running jobs of a user:

```
qselect -u $USER -s R | xargs qdel
```

Delete all queued jobs of a user:

```
qselect -u $USER -s Q | xargs qdel
```

5. Utilization Agreement

5.1. Policy

Users must be mindful that the HPC cluster is a shared resource. In particular, users must NOT use excessive resources on the HPC cluster that locks out others for large amounts of time. The following guidelines must be observed:

- **USAGE** All jobs are managed by the job scheduler which aims to keep both the fairness over all users and the efficiency of the cluster. If user has special requirements on the computing resources, contact IMTS. IMTS may allow a user or group of users to have sole access to the HPC cluster for a short time in special circumstances.
- **JOB** Users must be mindful of other resources which must be shared with other users, such as storage and memory. The set of processes running on a node should not consume more than the amount of average physical memory/core on that node. Users should make the consistent request on the CPU resources with what the job really uses.
- **LICENSING** Users must not use software which is licensed by another user or group without prior approval of the user or group which has paid for the licenses.
- **COMMUNITY** Users agree to be on the `hpc_users` mailing list and to read all emails sent to the above list. IMTS will communicate information via this list and will convene regular user group meetings. Users should attend such meetings where possible.
- **REPORTING** If a user has problems with the operation of the cluster or notices any failures of the hardware or software, please report any such problems to IMTS as soon as possible. IMTS appreciate any effort on detecting and reporting the problem.
- **LOGIN NODE** Users must submit all large jobs to the queue system from the login node and should avoid signing onto compute nodes. Small test jobs (less than a few minutes) may be run 1 at a time on the login node. The time-consuming or memory-consuming commands and programs must run as either batch or interactive jobs.
- **ADMINISTRATION** If cluster admins observe unreasonable user behavior, they will first contact the user by email, but if there is no response within an appropriate time, they may take all possible actions to stop the problem, such as deleting user jobs, deleting files, limiting job submission and executions etc.

5.2. Acknowledgements

When users publish their research work based on the UOW HPC Cluster, we would appreciate being mentioned in the *acknowledgment* section. Here is an example:

We would like to thank the University of Wollongong Information Management & Technology Services (IMTS) for computing time on the UOW High Performance Computing Cluster.

5.3. Contact Information

If you have any problems with or comments on the above document please contact IMTS. The following email addresses may be used.

- **hpc_users@uow.edu.au**: a mailing list comprising the HPC users. May be moderated.
- **hpc_admin@uow.edu.au**: the HPC administrators. Use the address for account requests or technical issues.

Appendices

A. Access the HPC cluster from Windows clients

If you want to access the HPC cluster from a Windows client, there are several tools such as ‘putty’ or ‘SSH Secure Shell Client’ you can use for this purpose.

A.1. Putty

Download ‘Putty’ and double click the downloaded file ‘putty.exe’ to start.

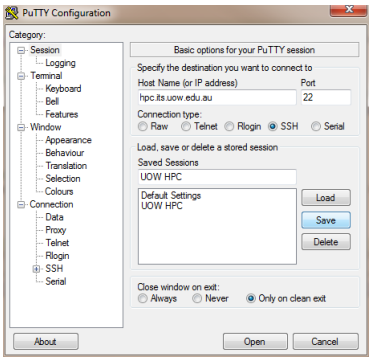

Figure A.1.: Start by selecting the Session tab and enter ‘hpc.its.uow.edu.au’ as the Host Name with SSH protocol selected. Fill in ‘UOW HPC’ in the Saved Sessions (or any other name you like) and click Save to save it. Click Open to open a login Window as shown below.

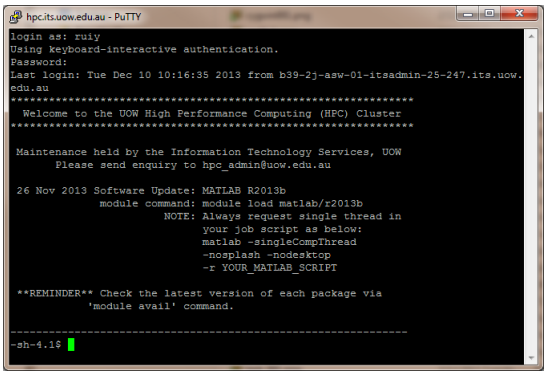
hpc.its.uow.edu.au - PuTTY
login as: ruiy
Using keyboard-interactive authentication.
Password:
Last login: Tue Dec 10 10:16:35 2013 from b39-2j-asw-01-itsadmin-25-247.its.uow.edu.au
Welcome to the UOW High Performance Computing (HPC) Cluster
Maintenance held by the Information Technology Services, UOW
Please send enquiry to hpc_admin@uow.edu.au
26 Nov 2013 Software Update: MATLAB R2013b
module command: module load matlab/r2013b
NOTE: Always request single thread in your job script as below:
matlab -singleCompThread -nosplash -nodesktop -r YOUR_MATLAB_SCRIPT
REMINDER Check the latest version of each package via 'module avail' command.
-sh-4.1$

Figure A.2.: Type in your username (‘ruiy’ herein as an example) and your password. Then you will see the Welcome and Notification message which means you have successfully logged in the UOW HPC cluster.

A.2. Configure ‘Putty’ with UOW proxy

If you cannot access the HPC cluster within the UOW, please consider to setup the proxy as below:

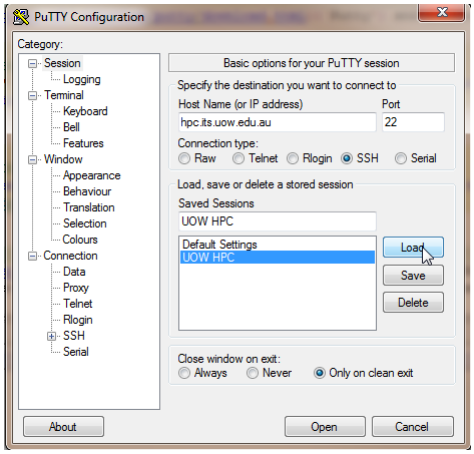

Figure A.3.: Select the Session of ‘hpc.its.upow.edu.au’ and click Load button.

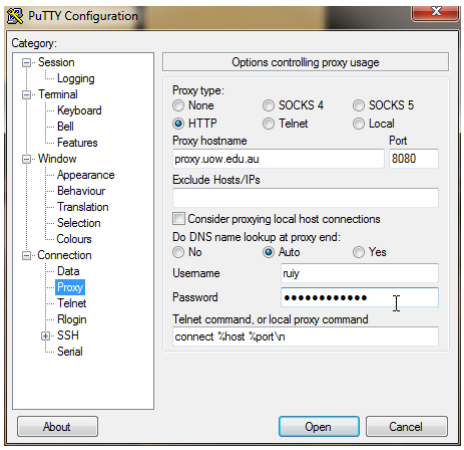

Figure A.4.: Click the Proxy tab in the left panel and fill in the ‘proxy.uow.edu.au’ as Proxy hostname, ‘8080’ as Port, your UOW account as Username and your UOW password as Password. Go back the the Session page to save it and click Open button to start login.

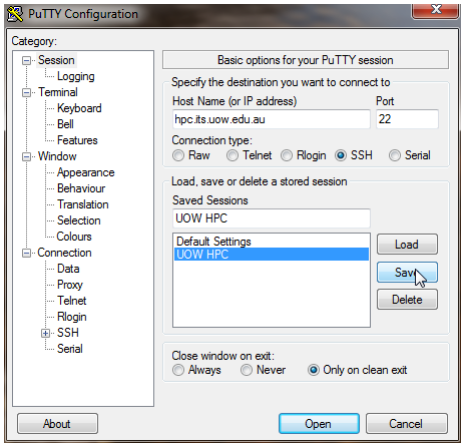

Figure A.5.: Go back to the Session page and save the proxy configuration by clicking Save button. Click Open button to start login.

A.3. SSH Secure Shell Client

Download 'SSH Secure Shell' and save it to an easily accessible place (your Windows desktop is a good choice). Start the installation by double-clicking the downloaded exe file in Windows Explorer. When the installation is complete, double click on the Desktop Icon to start the program.

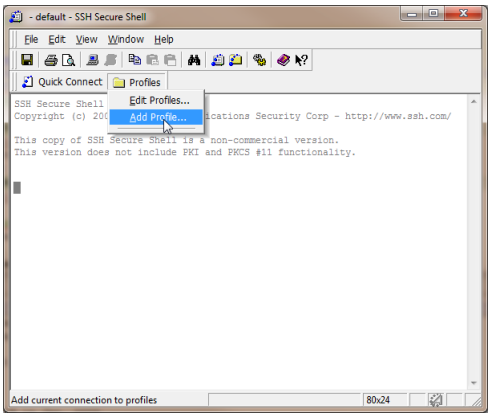
- default - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles
Edit Profiles...
Add Profile...
This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.
Add current connection to profiles
80x24

Figure A.6.: After installation, double click 'SSH Secure Shell Client' and click Profiles⇒Add Profile to add a profile 'UOW HPC'.

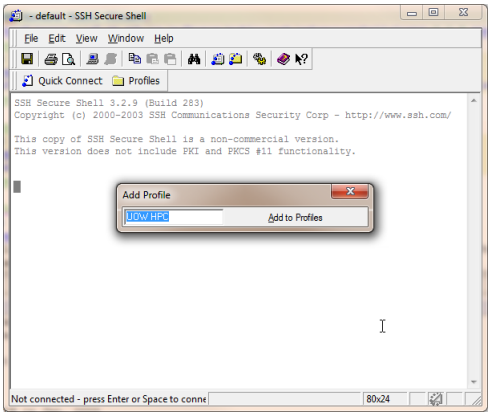
- default - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles
SSH Secure Shell 3.2.9 (Build 283)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/
This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.
Add Profile
UOW HPC
Add to Profiles
Not connected - press Enter or Space to conne
80x24

Figure A.7.: Add a profile named 'UoW HPC' and click Add to Profiles.

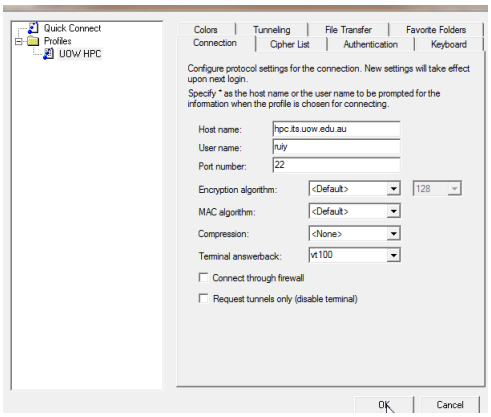

Figure A.8.: Click Profiles⇒Edit Profiles and select 'UOW HPC' from the Profiles session (Or any other name you like). Type in 'hpc.its.uow.edu.au' as the Host name and your username as User name. Click OK.

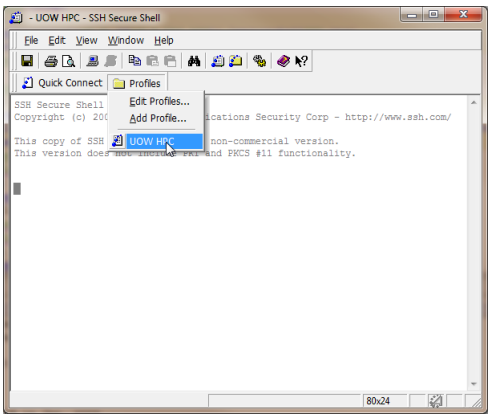

Figure A.9.: Click Profiles and select UOW HPC to log in the cluster.

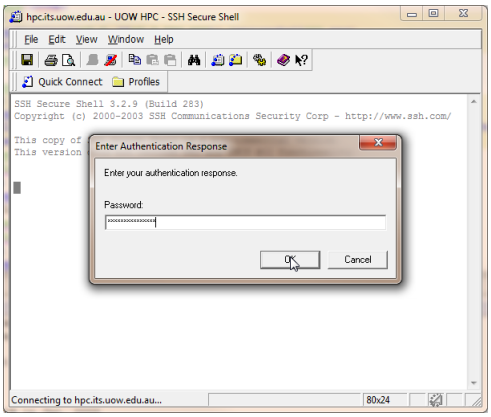

Figure A.10.: Type in the password and then click OK.

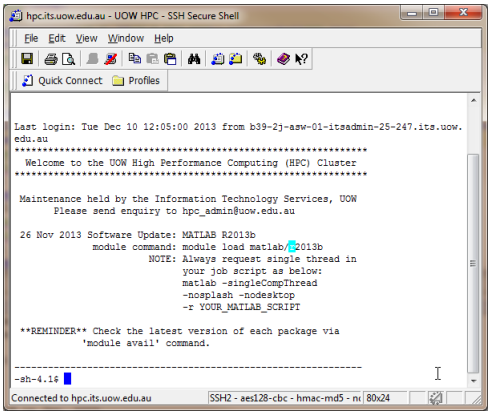
hpc.its.uow.edu.au - UOW HPC - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
Last login: Tue Dec 10 12:05:00 2013 from b39-2j-asw-01-itsadmin-25-247.its.uow.edu.au
Welcome to the UOW High Performance Computing (HPC) Cluster
Maintenance held by the Information Technology Services, UOW
Please send enquiry to hpc_admin@uow.edu.au
26 Nov 2013 Software Update: MATLAB R2013b
module command: module load matlab/R2013b
NOTE: Always request single thread in
your job script as below:
matlab -singleCompThread
-nosplash -nodesktop
-r YOUR_MATLAB_SCRIPT
REMINDER Check the latest version of each package via
'module avail' command.
-sh-4.1$
Connected to hpc.its.uow.edu.au
SSH2 - aes128-cbc - hmac-md5 - n
80x24

Figure A.11.: Now you have successfully logged into the UOW HPC cluster.

B. Enable Linux GUI applications using Xming

The X Window System is a system that allows graphical applications to be used on Unix-like operating systems instead of text-only applications. It is the foundation for Linux and Unix GUIs (Graphical User Interfaces). X (current version, X11) is defined by standards and contains standardized protocols. The X server is a process that runs on a computer with a bitmapped display, a keyboard, and a mouse. X clients are programs that send commands to open windows and draw in those windows. You can use either 'putty' or 'SSH Secure Shell' as the X client in conjunction with the X server 'Xming'.

B.1. Install Xming

1. Download 'Xming'
2. Double-click the Xming setup icon. The Xming Setup Wizard will start and the Setup Xming window will appear.
3. In the Setup Xming window, click NEXT to continue the installation.
4. When prompted for the installation location, choose the default path and Click NEXT.
5. When prompted for which components to install, accept the defaults. Click NEXT.
6. When prompted for the location for the shortcut, accept the default. Click NEXT.
7. When prompted for additional icons, select both the Xming and Xlaunch icons, if desired. Click NEXT.
8. Review the settings that you have selected. If no changes, click Install.
9. When the installation is complete, click Finish.
10. If the Windows Security Alert appears, your firewall is blocking all incoming traffic to your PC. To display on your screen, you need to select the Unblock option. This will add the necessary port to allow you to run X applications.
11. When Xming is running, you will see the Xming X symbol in your system tray on your Desktop.
12. To close Xming or to get more information about Xming, right-click on the Xming X symbol and choose from the drop down menu.
13. Xlaunch is a wizard that can be configured to start Xming sessions. Or you can simply start the Xming Server by click Xming icon.
14. You may also need to install Xming fonts to display characters correctly. Download 'Xming-fonts' and install it with default settings.

Next step, we need to configure the SSH clients such as 'Putty' or 'SSH Secure Shell Client' to run the X application.

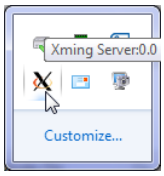

Figure B.1.: As soon as you see it, the X-server is running.

B.2. Configure ‘Putty’ with ‘Xming’

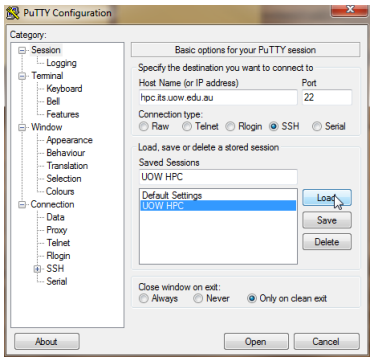

Figure B.2.: Open ‘putty’, select the ‘UOW HPC’ session and click Load.

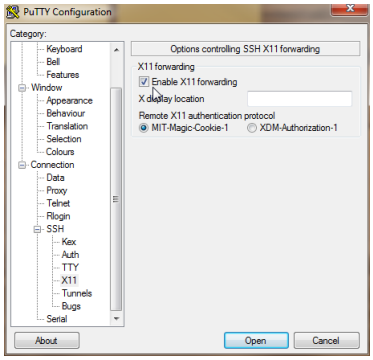

Figure B.3.: Go to SSH⇒X11 page and check Enable X11 forwarding.

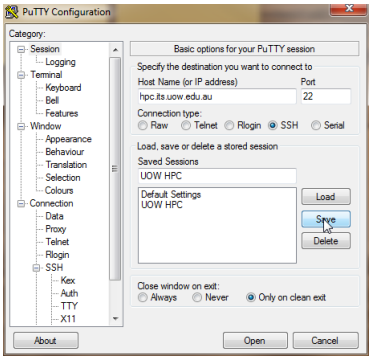

Figure B.4.: Go back to Session page and click Save to save the changes.

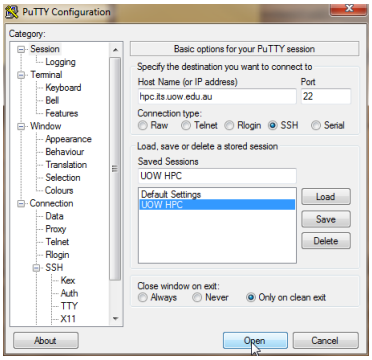

Figure B.5.: Select UOW HPC session to open the login window.

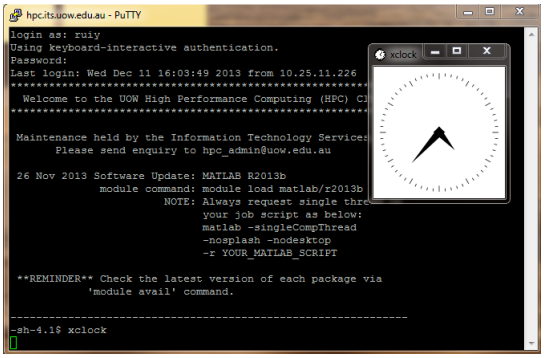
hpc.its.uow.edu.au - PuTTY
login as: ruiy
Using keyboard-interactive authentication.
Password:
Last login: Wed Dec 11 16:03:49 2013 from 10.25.11.226
Welcome to the UOW High Performance Computing (HPC) C
Maintenance held by the Information Technology Service
Please send enquiry to hpc_admin@uow.edu.au
26 Nov 2013 Software Update: MATLAB R2013b
module command: module load matlab/r2013b
NOTE: Always request single thre
your job script as below:
matlab -singleCompThread
-nosplash -nodesktop
-r YOUR_MATLAB_SCRIPT
REMINDER Check the latest version of each package via
'module avail' command.
-sh-4.1$ xclock
xclock

Figure B.6.: Select UOW HPC session to open the login window and type in your username ('ruiy' herein as an example) and password. After logging into the cluster, type '`xclock`' in the command line. If you see a clock displayed on your desktop, you are able to run other Linux X applications.

B.3. Configure ‘SSH Secure Shell Client’ with ‘Xming’

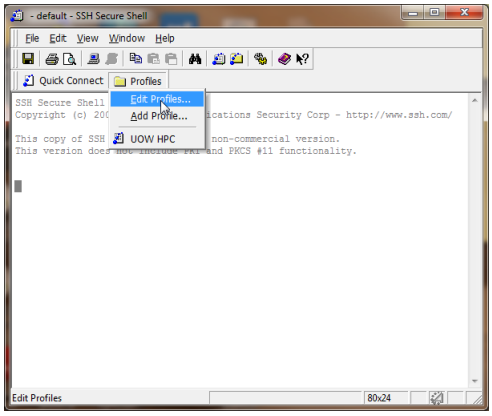

Figure B.7.: Start ‘SSH Secure Shell Client’, click Profiles and then Edit Profiles.

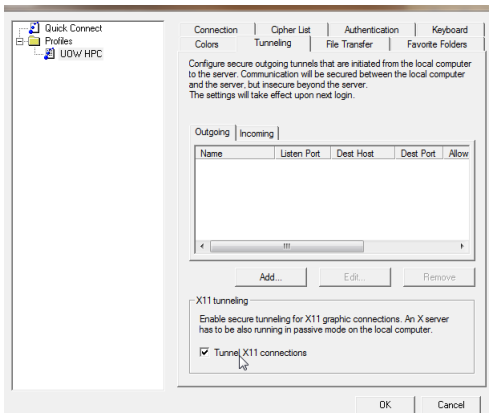

Figure B.8.: Enter ‘UOW HPC’ profile and the Tunneling tag, check Tunnel X11 connections and OK.

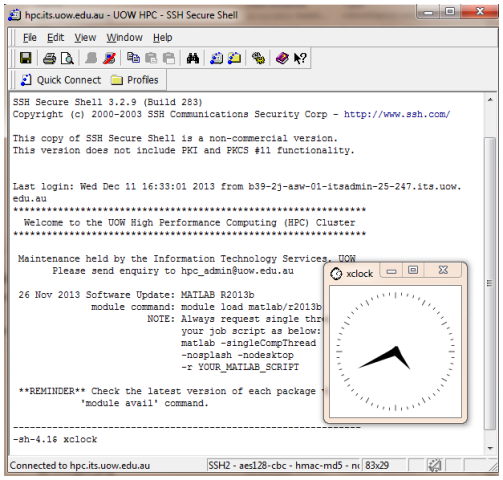
hpc.its.uow.edu.au - UOW HPC - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
SSH Secure Shell 3.2.9 (Build 283)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/
This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.
Last login: Wed Dec 11 16:33:01 2013 from b39-2j-asw-01-itsadmin-25-247.its.uow.edu.au
Welcome to the UOW High Performance Computing (HPC) Cluster
Maintenance held by the Information Technology Services, UOW
Please send enquiry to hpc_admin@uow.edu.au
26 Nov 2013 Software Update: MATLAB R2013b
module command: module load matlab/r2013b
NOTE: Always request single thr
your job script as below:
matlab -singleCompThread
-nosplash -nodesktop
-r YOUR_MATLAB_SCRIPT
REMINDER Check the latest version of each package
'module avail' command.
-sh-4.1$ xclock
xclock
Connected to hpc.its.uow.edu.au
SSH2 - aes128-cbc - hmac-md5 - n
83x29

Figure B.9.: Select UOW HPC profile to log into the cluster and type ‘xclock’ to test. If you see a clock displayed on your screen, you are successfully working in the X-Windows mode.

C. Transfer data between the Windows client and the HPC cluster

C.1. WinSCP

Download 'WinSCP' and install it using the default options. You will be asked to setup a connection.

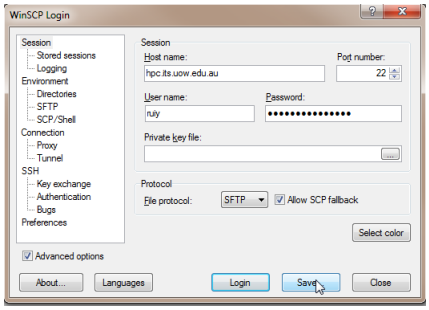

Figure C.1.: Type 'hpc.its.uow.edu.au' as the Host name and your USERNAME as the User name ('ruiy' herein as an example). You could choose to input your password here. Select 'Save' to continue.

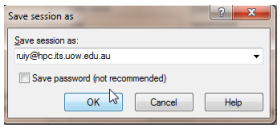

Figure C.2.: Save session. You could check Save password if you want the computer to remember your password.

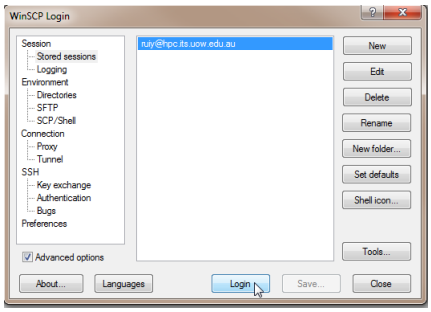

Figure C.3.: Select session ‘hpc.its.uow.edu.au’ and press Login to continue.

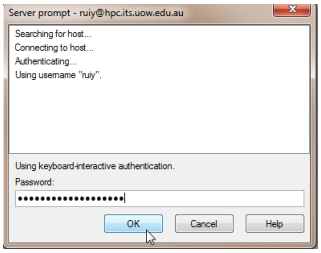

Figure C.4.: Enter your password here if you didn’t let WinSCP save it in the preceding step.

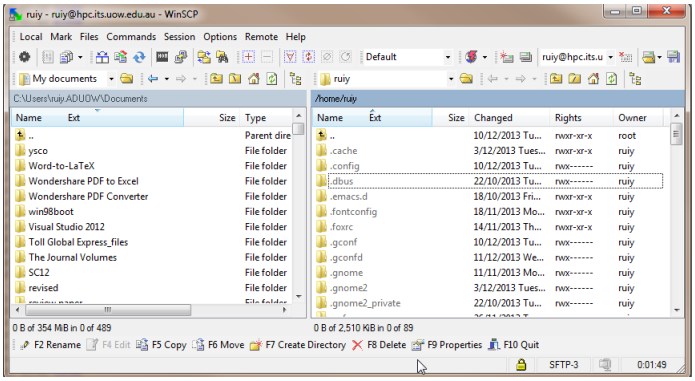
ruiy - ruiy@hpc.its.uow.edu.au - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
ruiy@hpc.its.u
My documents
ruiy
C:\Users\ruiy.ADUOW\Documents
/home/ruiy
Name Ext Size Type
Name Ext Size Changed Rights Owner
0 B of 354 MiB in 0 of 489
0 B of 2,510 KiB in 0 of 89
F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit
SFTP-3
0:01:49

Figure C.5.: Now you have logged into the HPC cluster successfully and you can drag and drop files to transfer data between the cluster and your desktop.

C.2. SSH Secure File Transfer Client

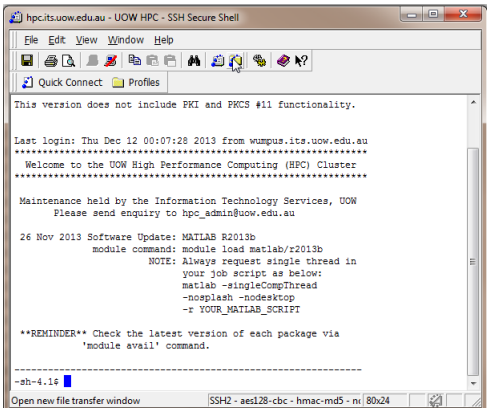
hpc.its.uow.edu.au - UOW HPC - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
This version does not include PKI and PKCS #11 functionality.
Last login: Thu Dec 12 00:07:28 2013 from wumpus.its.uow.edu.au
Welcome to the UOW High Performance Computing (HPC) Cluster
Maintenance held by the Information Technology Services, UOW
Please send enquiry to hpc_admin@uow.edu.au
26 Nov 2013 Software Update: MATLAB R2013b
module command: module load matlab/r2013b
NOTE: Always request single thread in
your job script as below:
matlab -singleCompThread
-nosplash -nodesktop
-r YOUR_MATLAB_SCRIPT
REMINDER Check the latest version of each package via
'module avail' command.
~sh-4.1$
Open new file transfer window
SSH2 - aes128-cbc - hmac-md5 - n
80x24

Figure C.6.: Open an SSH windows and connect to the UOW HPC cluster. Click the button under the mouse to open a 'SSH Secure File Transfer' windows as below.

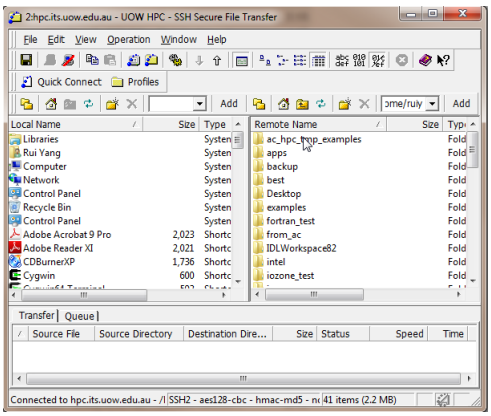

Figure C.7.: The file transfer interface opened with your local disk in the left column and the remote file system at the cluster in the right column. Now you can transfer files by 'drag and drop' to either column.

D. Transfer data at home (off the Campus)

As the UOW HPC cluster is located behind the university firewall, users can not transfer data directly between the HPC cluster and the client computer at home or off the campus. UOW staffs need to connect the UOW VPN firstly and then transfer the data by using the methods described in C. For users who can not access VPN, ssh tunnel via the UOW gateway server '`wumpus.uow.edu.au`' can be used to transfer data at home.

D.1. Windows OS

For client computer with Windows OS, first install WinSCP and then follow steps below:

1. Create a new session connecting to '`hpc.its.uow.edu.au`'.
2. Edit the session and select the Tunnel from the left panel, click Connect through SSH tunnel, and fill in '`wumpus.uow.edu.au`' as the hostname and your UOW username as the username.
3. Save above and connect it.

Now you should be able to transfer data between your home computer and the HPC cluster.

D.2. Linux or Mac OS

On a computer with Linux or Mac OS, there is no need to install additional packages. The OS implemented `scp` command can be used to transfer data between the home computer and the HPC cluster.

D.2.1. Transfer data from your home computer to the HPC cluster

1. From your home computer, open a command line terminal and set the forwarding port with any valid number (i.e. 1234 for example) as below:

   ```
   $ ssh -L 1234:hpc.its.uow.edu.au:22 YOUR_UOW_USER_NAME@wumpus.uow.edu.au
   ```

   After typing in your password, leave the above terminal open.
2. Open another command line terminal and type:

   ```
   scp -P 1234 -r SOURCE_DIRECTORY YOUR_UOW_USER_NAME@127.0.0.1:TARGET_DIRECTORY
   ```

   This command will copy the whole SOURCE_DIRECTORY at your home computer to the TARGET_DIRECTORY (absolute path) at the HPC cluster.

D.2.2. Transfer data from the HPC cluster to your home computer

1. From your home computer, open a command line terminal and set the forwarding port with any valid number (i.e. 1234 for example) as below:

   ```
   $ ssh -L 1234:hpc.its.uow.edu.au:22 YOUR_UOW_USER_NAME@wumpus.uow.edu.au
   ```

   After typing in your password, leave the above terminal open.

2. Open another command line terminal and type:

   ```
   scp -P 1234 -r YOUR_UOW_USER_NAME@127.0.0.1:SOURCE_DIRECTORY TARGET_DIRECTORY
   ```

   This command will copy the whole SOURCE_DIRECTORY (absolute path) at the HPC cluster to the TARGET_DIRECTORY at your home computer.

E. Selected Linux Commands

Table E.1.: Selected Linux commands description

Commands	Description
	System information
`uname -a`	Show kernel version and system architecture
`hostname`	Show host server name
`hostname -i`	Show host server IP address
	file searching
`ls; ls -l; ls -lr; ls -lrt`	List files; with more information; by filename; by date
`which cmd`	Show full path name of command *cmd*
`find . -name arg`	Search the file named *arg* under the current directory
	File management
`cp file1 file2`	copy file *file1* to *file2*
`cp -r dir1 dir2`	copy directory *dir1* to *dir2*
`rm file1`	remove(delete) file *file1*
`rm -r dir1`	remove(delete) directory *dir1*
`mv file1 dir2/file2`	move and/or rename file *file1* to *dir2/file2*
	Directory navigation
`pwd`	Show current directory
`mkdir dir`	Create a directory 'dir'
`cd; cd -; cd dir`	Goto $HOME directory; previous directory; directory *dir*
	Disk space
`ls -lSrh`	Show files by size, biggest last with human friendly unit
`df -h`	Show free space on mounted filesystems
`du -sh *`	Show size of files and subdirectories of the current directory
	Manipulating files
`tail -f file`	Monitor messages in a log file (Ctrl+C to exit)
`more file`	Displays the contents of the file one screen at a time, use 'space' for next page
`cat file`	Lists all contents of files to the screen
`grep args file`	Search argument *args* in *file*
	Archives and compression
`tar -cvf dir.tar dir`	Make archive of *dir* as *dir.tar*
`tar -xvf dir.tar`	Extract files from the archive of *dir.tar*
`gzip file`	Compress *file* in a file *file*.gz
`gunzip file.gz`	Extract *file*.gz to file *file*
	On-line help
`man cmd`	Show the help information for command *cmd*

Continued on next page

Table E.1 – continued from previous page

Commands	Description
`apropos` *subject*	Displays a list of all topics in the man pages that are related to the subject of a query, i.e. *subject*. It is particularly useful when searching for commands without knowing their exact names.

F. Software Guide

NOTE: Please add '`source /etc/profile.d/modules.sh`' before any module command in your job script when submit to the queue system.

F.1. Parallel Programming Libraries/Tools

F.1.1. Intel MPI

Intel MPI Library focuses on making applications perform better on Intel architecture-based clusters-implementing the high performance Message Passing Interface specification on multiple fabrics. It enables you to quickly deliver maximum end user performance even if you change or upgrade to new interconnects, without requiring changes to the software or operating environment.

Version History

- 2015/02/18 version 2015sp2;
- 2014/09/11 version 2015;
- 2013/10/22 version 4.1.1.036(2013sp1);

How to Use

Intel MPI is integrated into the Intel Cluster Studio XE. Put one of the following lines in your job script or in the command line to use the Intel MPI library:

```
module load intel_mpi/2015_sp2
module load intel_mpi/2015
module load intel_mpi/2013_sp1
```

F.1.2. MPICH

MPICH is a high performance and widely portable implementation of the Message Passing Interface (MPI) standard. The goals of MPICH are:

- to provide an MPI implementation that efficiently supports different computation and communication platforms including commodity clusters (desktop systems, shared memory systems, multicore architectures), high-speed networks and proprietary high-end computing systems (Blue Gene, Cray)
- to enable cutting-edge research in MPI through an easy-to-extend modular framework for other derived implementations

Version History

- 2014/03/10 version 1.5 GCC build;
- 2013/10/24 version 3.0.4 Open64 compiler build;
- 2013/10/22 version 3.0.4 GCC build;
- 2013/10/22 version 3.0.4 PGI compiler build;
- 2013/10/22 version 3.0.4 Intel compiler build;

How to Use

Put one of the following lines in your job script or in the command line to use MPICH for different compiler builds.

- GCC compiler build

  ```
  module load mpich/3.0.4_gcc
  ```

  or

  ```
  module load mpich/1.5_gcc
  ```

- Intel compiler build

  ```
  module load mpich/3.0.4_itl
  ```

- PGI compiler build

  ```
  module load mpich/3.0.4_pgi
  ```

- Open64 compiler build

  ```
  module load mpich/3.0.4_open64
  ```

F.1.3. OpenMPI

OpenMPI is a high performance message passing library. The Open MPI Project is an open source MPI-2 implementation that is developed and maintained by a consortium of academic, research, and industry partners. Open MPI is therefore able to combine the expertise, technologies, and resources from all across the High Performance Computing community in order to build the best MPI library available. Open MPI offers advantages for system and software vendors, application developers and computer science researchers.

Version History

- 2013/10/25 version 1.6.5 Open64 compiler build;
- 2013/10/23 version 1.4.4 GCC build;
- 2013/10/22 version 1.4.4 PGI compiler build;
- 2013/10/22 version 1.6.5 GCC build;
- 2013/10/22 version 1.6.5 PGI compiler build;
- 2013/10/22 version 1.6.5 Intel compiler build;

How to Use

Put one of the following lines in your job script or in the command line to use OpenMPI for different compiler builds.

- GCC compiler build version 1.4.4

```
module load openmpi/1.4.4_gcc
```

- PGI compiler build version 1.4.4

```
module load openmpi/1.4.4_pgi
```

- GCC compiler build version 1.6.5

```
module load openmpi/1.6.5_gcc
```

- PGI compiler build version 1.6.5

```
module load openmpi/1.6.5_pgi
```

- Intel compiler build version 1.6.5

```
module load openmpi/1.6.5_itl
```

- Open64 compiler build version 1.6.5

```
module load openmpi/1.6.5_open64
```

OpenMPI is **RECOMMENDED** as your default MPI environment.

F.2. Compilers & Building Tools

F.2.1. CMake

The cross-platform, open-source build system CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and work spaces that can be used in the compiler environment of your choice.

Version History

- 2013/10/23 version 2.8.12

How to Use

Put one of the following lines in your job script or in the command line to use different versions of CMake.

- version 2.8.12

```
module load cmake/2.8.12
```

F.2.2. GNU Compiler Collection (GCC)

The GNU Compiler Collection (GCC) includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgcj etc.). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100% free software, free in the sense that it respects the users freedom.

Version History

- 2014/11/07 version 4.9.2
- 2013/10/22 version 4.7.1
- 2013/10/19 version 4.8.2

How to Use

The system default GCC version is 4.4.7, which can be used directly. Put one of the following lines in your job script or in the command line to use different versions of GCC.

- version 4.7.1

```
module load gcc/4.7.1
```

- version 4.8.2

```
module load gcc/4.8.2
```

- version 4.9.2

```
module load gcc/4.9.2
```

F.2.3. Intel C, C++&Fortran Compiler

Intel Composer XE delivers outstanding performance for your applications as they run on systems using Inte Core or Xeon processors, including Intel Xeon Phi coprocessors, and IA-compatible processors. It combines all the tools from Intel C++ Composer XE with those from Intel Fortran Composer XE.

Version History

- 2015/02/18 version 2015sp2
- 2014/09/11 version 2015
- 2013/10/22 version 2013 SP1

How to Use

Put one of the following lines in your job script or run it in the command line to use different versions of Intel Composer XE.

- version 2015sp2

```
module load intel_ics/2015_sp2
```

- version 2015

```
module load intel_ics/2015
```

- version 2013 SP1

```
module load intel_ics/2013_sp1
```

F.2.4. Open64

Open64 has been well-recognized as an industrial-strength production compiler. It is the final result of research contributions from a number of compiler groups around the world. Open64 includes advanced interprocedural optimizations, loop nest optimizations, global scalar optimizations, and code generation with advanced global register allocation and software pipelining.

Version History

- 2013/10/18 version 5.0

How to Use

Put one of the following lines in your job script or in the command line to use different versions of Open64.

- version 5.0

```
module load open64/5.0
```

F.2.5. PGI Fortran/C/C++ Compiler

PGI Workstation is a scientific and engineering compilers and tools product. It combines PGI Fortran Workstation and PGI C/C++ Workstation. PGI Fortran Workstation for Linux includes The Portland Groups native parallelizing/optimizing Fortran 2003, FORTRAN 77 and HPF compilers. It provides the features, quality, and reliability necessary for developing and maintaining advanced scientific and technical applications. PGI C/C++ Workstation includes The Portland Groups native parallelizing/optimizing OpenMP C++ and ANSI C compilers. The C++ compiler closely tracks the proposed ANSI standard and is compatible with cfront versions 2 and 3. All C++ functions are compatible with Fortran and C functions, so you can compose programs from components written in all three languages. PGI Workstation includes the OpenMP and MPI enabled PGDBG parallel debugger and PGPROF performance profiler that can debug and profile up to eight local MPI processes. PGI Workstation also includes a precompiled MPICH message passing library.

Version History

- 2015/03/05 version 15.3
- 2014/08/13 version 14.7
- 2013/10/21 version 13.9

How to Use

Put one of the following lines in your job script or in the command line to use different versions of PGI compiler.

- version 13.9

```
module load pgi/13.9
```

- version 14.7

```
module load pgi/14.7
```

- version 15.3

```
module load pgi/15.3
```

F.3. Scripting Languages

F.3.1. IPython

IPython is a command shell for interactive computing in multiple programming languages, especially focused on the Python programming language, that offers enhanced introspection, rich media, additional shell syntax, tab completion, and rich history.

Version History

- 2013/11/11 version 1.1.0 based on Python 2.6.6

How to Use

Put one of the following lines in your job script or in the command line to use different versions of iPython compiler.

- version 1.1.0

```
module load ipython/1.1.0
```

F.3.2. Java

Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers ”write once, run anywhere” (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture.
The system default version of Java is 1.7.0_13.

F.3.3. Perl

Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages.

Version History

- 2015/02/24 Perl 5.20.2

How to Use

The system default version of Perl is 5.10.1 which can be used directly. To use version 5.20.2, load the module as below

```
module load perl/5.20.2
```

F.3.4. Python

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. The language provides constructs intended to enable clear programs on both a small and large scale.

Version History

- 2014/05/20 Python 2.7.6

How to Use

The system default version of Python is 2.6.6 which can be used directly. To use version 2.7.6, load the module as below

```
module load python/2.7.6
```

F.4. Code Development Utilities

F.4.1. Eclipse for Parallel Application Developers

Eclipse for Parallel Application Developers (PAD) is an IDE for Parallel Application Developers. Includes the C/C++ IDE, plus tools for Fortran, UPC, MPI, a parallel debugger, etc.

Version History

- 2013/10/22 Eclipse PAD Kepler_SR1
- 2014/03/03 Eclipse PAD Kepler_SR2

How to Use

To use PAD, load the module as below

```
module load eclipse_pad/kepler_sr1
```

or

```
module load eclipse_pad/kepler_sr2
```

This build has implemented the Intel Fortran/C/C++ compilers. You must enable the X-window environment at the client side to use the Eclipse GUI.
For more details on using Eclipse, please read its online documents.

F.5. Math Libraries

F.5.1. AMD Core Math Library (ACML)

ACML provides a free set of thoroughly optimized and threaded math routines for HPC, scientific, engineering and related compute-intensive applications. ACML consists of the following main components:

- A full implementation of Level 1, 2 and 3 BLAS and LAPACK, with key routines optimized for high performance on AMD Opteron processors.
- A comprehensive suite of FFTs in single-, double-, single and double-complex data types.
- Fast scalar, vector, and array math transcendental library routines optimized for high performance on AMD Opteron processors.
- Random Number Generators in both single- and double-precision.

Version History

- 2014/11/21 ACML 6.1.0
- 2013/10/23 ACML 5.3.1

How to Use

ACML support multiple compilers such as gfortran, PGI, Open64 and Intel.

Number of threads	Compilers	ACML install directory
Single thread	PGI pgf77/pgf90/pgcc	pgi64
	PGI pgf77/pgf90/pgcc fma4	pgi64_fma64
	GNU gfortran/gcc or compat.	gfortran64
	GNU gfortran/gcc fma4	gfortran_64_fma4
	Open64 openf95/opencc	open64_64
	Open64 openf95/opencc fma4	open64_64_fma4
	Intel Fortran	ifort64
	Intel Fortran fma4	ifort64_fma4
Multiple threads	PGI pgf77/pgf90/pgcc	pgi64_mp
	PGI pgf77/pgf90/pgcc fma4	pgi64_fma64_mp
	GNU gfortran/gcc or compat.	gfortran64_mp
	GNU gfortran/gcc fma4	gfortran_64_fma4_mp
	Open64 openf95/opencc	open64_64_mp
	Open64 openf95/opencc fma4	open64_64_fma4_mp
	Intel Fortran	ifort64_mp
	Intel Fortran fma4	ifort64_fma4_mp

Table F.1.: ACML libraries locations (under /hpc/software/package/acml/5.3.1/)

Specify the library location to one of the above ACML installation directories when building your code. For example, the command:

```
gfortran -m64 driver.f \
-L/hpc/software/package/acml/5.3.1/gfortran64/lib -lacml
```

can be used to compile the program `driver.f` and link it to the ACML.
The command

```
gfortran -m64 -mavx -mfma4 driver.f \
-L/hpc/software/package/acml/5.3.1/gfortran64_fma4/lib -lacml
```

will compile and link a 64-bit program with the 64-bit FMA4 ACML. The Fortran module `driver.f` will be compiled using AVX and FMA4 instructions where possible.
Please refer to the ACML Manual for further details on how to use it.

F.5.2. Automatically Tuned Linear Algebra Software (ATLAS)

The ATLAS project is an ongoing research effort focusing on applying empirical techniques in order to provide portable performance. At present, it provides C and Fortran77 interfaces to a portably efficient BLAS implementation, as well as a few routines from LAPACK. The ATLAS installed at the HPC cluster provides a full LAPACK package. ATLAS supports gfortran compiler.

Version History

- 2013/11/03 ATLAS 3.11.17 GCC build

How to Use

To use ATLAS static library, select libraries as below

```
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libatlas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libcblas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libf77blas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/liblapack.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libptcblas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libptf77blas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libptlapack.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libsatlas.so
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libtatlas.so
```

as the compiling flags.

F.5.3. Basic Linear Algebra Communication Subprograms (BLACS)

The BLACS project is an ongoing investigation whose purpose is to create a linear algebra oriented message passing interface that may be implemented efficiently and uniformly across a large range of distributed memory platforms.

Version History

- 2013/12/12 OpenMPI 1.6.5 GCC build

How to Use

To use the MPI BLACS static library, include

```
/hpc/software/package/blacs/gcc/blacsF77init_MPI-LINUX-0.a
/hpc/software/package/blacs/gcc/blacs_MPI-LINUX-0.a
/hpc/software/package/blacs/gcc/blacsCinit_MPI-LINUX-0.a
```

as your compiling flag within the OpenMPI environment loaded by

```
module load openmpi/1.6.5_gcc
```

F.5.4. Basic Linear Algebra Subroutines (BLAS)

The Basic Linear Algebra Subroutines (BLAS) are fundamental to many linear algebra software packages. The 3 levels of BLAS corresponding to the following operations respectively:

```
Level 1 BLAS: vector*vector
Level 2 BLAS: vector*matrix
Level 3 BLAS: matrix*matrix
```

The BLAS routines can be found in several packages such as Intel MKL, Atlas etc.

Version History

- 2015/03/05 BLAS 3.5.0

How to Use

To use BLAS dynamic library, include

```
-L/hpc/software/package/lapack/3.4.2_gcc/lib -lblas
```

as the compiling flags.
To use BLAS static library, include

```
/hpc/software/package/blas/3.5.0_gcc/blas_LINUX.a
/hpc/software/package/blas/3.5.0_gcc/cblas_LINUX.a
```

as the compiling flags.

F.5.5. Boost

Boost provides free peer-reviewed portable C++ source libraries.

Version History

- 2013/12/17 BOOST 1.55.0

How to Use

Load the Boost module as below:

```
module load boost/1.55.0
```

When building the code, specify the following head file and library directories:

```
/hpc/software/package/boost/1.55.0/include
```

and

```
/hpc/software/package/boost/1.55.0/lib/libboost_thread.a
```

respectively. The Boost MPI library is built based on `MPICH 3.0.4_gcc`.
To compile anything in Boost, you need a directory containing the boost/ subdirectory in your `#include` path. Since all of Boost's header files have the `.hpp` extension, and live in the `boost/` subdirectory of the boost root, your Boost `#include` directives will look like:

```
#include <boost/whatever.hpp>
```

or

```
#include "boost/whatever.hpp"
```

depending on your preference regarding the use of angle bracket includes.
For example, to build the following example code by using Boost thread library, i.e. '`example.cpp`'

```
#include <iostream>
#include <boost/thread.hpp>p
#include <boost/date_time.hpp>

void workerFunc()
{
    boost::posix_time::seconds workTime(3);
    std::cout << "Worker:␣running" << std::endl;
    // Pretend to do something useful...
    boost::this_thread::sleep(workTime);
    std::cout << "Worker:␣finished" << std::endl;
}

int main(int argc, char* argv[])
{
    std::cout << "main:␣startup" << std::endl;
    boost::thread workerThread(workerFunc);
    std::cout << "main:␣waiting␣for␣thread" << std::endl;
    workerThread.join();

    std::cout << "main:␣done" << std::endl;
    return 0;
}
```

invoke the Boost include and library path as below:

```
c++ ./example.cpp -o example -pthread \
-I/hpc/software/package/boost/1.55.0/include \
-L/hpc/software/package/boost/1.55.0/lib -lboost_thread
```

Please note, the '`-pthread`' flag is not a Boost library but a system library to warrant the Boost thread library working. You don't need to invoke it when linking other Boost libraries.

Also remember to set the library path to the system linking path `LD_LIBRARY_PATH` when run the program linking with the shared library:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/hpc/software/package/boost/1.55.0/lib
```

Check the BOOST Manual for further details on how to use it.

F.5.6. FFTW

FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DCT/DST).

Version History

- 2014/11/21 GCC Compiler build version 3.3.4
- 2013/11/14 Intel Compiler build version 2.1.5
- 2013/11/11 GCC Compiler build version 3.3.3
- 2013/11/08 Intel Compiler build version 3.3.3
- 2013/11/07 PGI Compiler build version 3.3.3

How to Use

Put one of the following lines in your job script or in the command line to use different versions of FFTW.

- Intel Compiler build version 2.1.5

```
module load fftw/2.1.5_itl
```

- PGI Compiler build version 3.3.3

```
module load fftw/3.3.3_pgi
```

- Intel Compiler build version 3.3.3

```
module load fftw/3.3.3_itl
```

- GCC Compiler build version 3.3.3

```
module load fftw/3.3.3_gcc
```

- GCC Compiler build version 3.3.4

```
module load fftw/3.3.4_gcc_4.8.2
```

F.5.7. The GNU Multiple Precision Arithmetic Library (GMP)

GMP is a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating point numbers. There is no practical limit to the precision except the ones implied by the available memory in the machine GMP runs on. GMP has a rich set of functions, and the functions have a regular interface. The main target applications for GMP are cryptography applications and research, algebra systems, Internet security applications, computational algebra research, etc.

Version History

- 2014/02/19 GMP 5.1.3
- 2013/10/05 GMP 4.3.2

How to Use

GMP is complied using gcc and g++. To use GMP, include one of the following static libraries

```
/hpc/software/package/gmp/4.3.2/lib/libgmp.a
/hpc/software/package/gmp/5.1.3/lib/libgmp.a
```

or link one of the following shared libraries

```
/hpc/software/package/gmp/4.3.2/lib/libgmp.so
/hpc/software/package/gmp/5.1.3/lib/libgmp.so
```

F.5.8. The GNU Scientific Library (GSL)

The GNU Scientific Library (GSL) is a numerical library for C and C++ programmers. It is free software under the GNU General Public License.

The library provides a wide range of mathematical routines such as random number generators, special functions and least-squares fitting. There are over 1000 functions in total with an extensive test suite.

The complete range of subject areas covered by the library includes,

- Complex Numbers Roots of Polynomials
- Special Functions Vectors and Matrices
- Permutations Sortingp
- BLAS Support Linear Algebra
- Eigensystems Fast Fourier Transforms
- Quadrature Random Numbers
- Quasi-Random Sequences Random Distributions
- Statistics Histograms
- N-Tuples Monte Carlo Integration
- Simulated Annealing Differential Equations
- Interpolation Numerical Differentiation
- Chebyshev Approximation Series Acceleration
- Discrete Hankel Transforms Root-Finding
- Minimization Least-Squares Fitting
- Physical Constants IEEE Floating-Point
- Discrete Wavelet Transforms Basis splines

Unlike the licenses of proprietary numerical libraries the license of GSL does not restrict scientific cooperation. It allows you to share your programs freely with others.

Version History

- 2013/11/12 version 1.16 GCC build

How to Use

Load the module of gsl as below:

```
module load gsl/1.16_gcc
```

To build the code using gsl, add the following flags:

```
-L/hpc/software/package/gsl/1.16/lib -lgsl -lgslcblas \
-I/hpc/software/package/gsl/1.16/include
```

An example code can be find at

```
/hpc/tmp/examples/gsl/1.16_gcc
```

F.5.9. Intel Math Kernel Library (IMKL)

IMKL is Intel's math library optimised for the Intel compiler. It includes the BLAS and LAPACK routines. The IMKL library also includes FFT routines, some sparse solvers and a vector statistical library which provides random number generators.

Version History

- 2015/02/18 version 11.2 for Intel ICS 2015sp2
- 2014/09/11 version 11.2 for Intel ICS 2015
- 2013/10/22 version 11.1

How to Use

Put one of the following lines in your job script or in the command line to use different versions of Intel MKL.

- version 11.2

  ```
  module load intel_mkl/2015_11.2
  ```

  or

  ```
  module load intel_mkl/2015sp2_11.2
  ```

- version 11.1

  ```
  module load intel_mkl/11.1
  ```

To learn more on how to use IMKL, please visit IMKL site.

F.5.10. Linear Algebra PACKage (LAPACK)

Linear Algebra PACKage (LAPACK) is a collection of routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. The associated matrix factorizations (LU, Cholesky, QR, SVD, Schur, generalized Schur) are also provided, as are related computations such as reordering of the Schur factorizations and estimating condition numbers. Dense and banded matrices are handled, but not general sparse matrices. In all areas, similar functionality is provided for real and complex matrices, in both single and double precision. This is a `GCC` version.

Version History

- 2013/11/11 GCC Compiler build version 3.4.2
- 2014/04/11 GCC Compiler build version 3.5.0

How to Use

To use LAPACK static library, include

```
-L/hpc/software/package/lapack/3.4.2_gcc/lib -llapack -lblas
```

or

```
-L/hpc/software/package/lapack/3.5.0_gcc/lib -llapack -lrefblas
```

as the compiling flags.

F.5.11. Multiple-Precision Floating-point with correct Rounding(MPFR)

The MPFR library is a C library for multiple-precision floating-point computations with correct rounding. MPFR is based on the GMP multiple-precision library. The main goal of MPFR is to provide a library for multiple-precision floating-point computation which is both efficient and has a well-defined semantics. It copies the good ideas from the ANSI/IEEE-754 standard for double-precision floating-point arithmetic (53-bit significand).

Version History

- 2014/02/19 MPFR 3.1.2
- 2013/10/18 MPFR 2.4.2

How to Use

MPFR is complied using gcc and g++. To use MPFR, include one of the following static libraries

```
/hpc/software/package/mpfr/2.4.2/lib/libmpfr.a
/hpc/software/package/mpfr/3.1.2/lib/libmpfr.a
```

or link one of the following shared libraries

```
/hpc/software/package/mpfr/2.4.2/lib/libmpfr.so
/hpc/software/package/mpfr/3.1.2/lib/libmpfr.so
```

F.5.12. NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Version History

- 2015/01/28 NumPy 1.9.1 based on Python 2.6.6
- 2014/06/17 NumPy 1.8.1 based on Python 2.6.6
- 2014/06/13 NumPy 1.8.1 based on Python 2.7.6
- 2013/11/11 NumPy 1.8.0 based on Python 2.6.6

How to Use

Load one of the following modules:

```
module load numpy/1.8.0
```

or

```
module load numpy/1.8.1
```

or

```
module load numpy/1.9.1
```

for python 2.6;

```
module load numpy/1.8.1\_py27
```

for python 2.7; For more details on using NumPy, please read tutorial.

F.5.13. Scalable LAPACK (ScaLAPACK)

TheScaLAPACK library includes a subset of LAPACK routines redesigned for distributed memory MIMD parallel computers. It is currently written in a SPMD (Single Program Multiple Data) style using explicit message passing for interprocessor communication. It assumes matrices are laid out in a two-dimensional block cyclic decomposition. It is designed for heterogeneous computing and is portable on any computer that supports MPI (or PVM).

Like LAPACK, the ScaLAPACK routines are based on block-partitioned algorithms in order to minimize the frequency of data movement between different levels of the memory hierarchy. (For such machines, the memory hierarchy includes the off-processor memory of other processors, in addition to the hierarchy of registers, cache, and local memory on each processor.) The fundamental building blocks of the ScaLAPACK library are distributed memory versions (PBLAS) of the Level 1, 2 and 3 BLAS, and a set of Basic Linear Algebra Communication Subprograms (BLACS) for communication tasks that arise frequently in parallel linear algebra computations. In the ScaLAPACK routines, all interprocessor communication occurs within the PBLAS and the BLACS. One of the design goals of ScaLAPACK was to have the ScaLAPACK routines resemble their LAPACK equivalents as much as possible. This is a OpenMPI version.

Version History

- 2013/11/11 ScalLAPACK 2.0.2 GCC 4.8.2+ACML build
- 2013/11/07 ScalLAPACK 2.0.2 GCC 4.8.2+ATLAS build

How to Use

First load the necessary modules as below

```
module load gcc/4.8.2
module load openmpi/1.6.5_gcc
```

Then include the following libaries when building your code:

- ScalLAPACK 2.0.2 GCC 4.8.2+ACML build

```
/hpc/software/package/scalapack/2.0.2_gcc_4.8.2_acml/libscalapack.a
/hpc/software/package/acml/5.3.1/gfortran64_fma4/lib/libacml.a
```

- ScalLAPACK 2.0.2 GCC 4.8.2+ATLAS build

```
/hpc/software/package/scalapack/2.0.2_gcc_4.8.2_atlas/libscalapack.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libatlas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libcblas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libf77blas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/liblapack.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libptcblas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libptf77blas.a
/hpc/software/package/ATLAS/3.11.17_gcc_fma4/lib/libptlapack.a
```

F.5.14. SciPy

SciPy refers to several related but distinct entities:

- The SciPy Stack, a collection of open source software for scientific computing in Python, and particularly a specified set of core packages.
- The community of people who use and develop this stack.
- Several conferences dedicated to scientific computing in Python - SciPy, EuroSciPy and SciPy.in.
- The SciPy library, one component of the SciPy stack, providing many numerical routines.

Version History

- 2015/01/28 SciPy 0.15.1 based on Python 2.6.6
- 2013/11/11 SciPy 0.14.0 based on Python 2.7.6
- 2013/11/11 SciPy 0.13.0 based on Python 2.6.6

How to Use

Select the appropriate module as below based on the Python versions:

- Python 2.6

```
module load scipy/0.15.1
```

or

```
module load scipy/0.13.0
```

- Python 2.7

```
module load scipy/0.14.0_py27
```

For more details on using SciPy, please read its online documents.

F.6. Debuggers, Profilers and Simulators

F.6.1. HPCToolkit

HPCToolkit is an integrated suite of tools for measurement and analysis of program performance on computers ranging from multicore desktop systems to the nation's largest supercomputers. By using statistical sampling of timers and hardware performance counters, HPCToolkit collects accurate measurements of a program's work, resource consumption, and inefficiency and attributes them to the full calling context in which they occur. HPCToolkit works with multilingual, fully optimized applications that are statically or dynamically linked. Since HPCToolkit uses sampling, measurement has low overhead (1-5%) and scales to large parallel systems. HPCToolkit's presentation tools enable rapid analysis of a program's execution costs, inefficiency, and scaling characteristics both within and across nodes of a parallel system. HPCToolkit supports measurement and analysis of serial codes, threaded codes (e.g. pthreads, OpenMP), MPI, and hybrid (MPI+threads) parallel codes.

Version History

- 2015/03/31 HPCToolkit 3.5.2

How to Use

To use valgrind, load one of the valgrind modules as below

```
module load hpctoolkit/3.5.2
```

For more details on using Valgrind, please read its online documents.

F.6.2. TAU

TAU Performance System is a portable profiling and tracing toolkit for performance analysis of parallel programs written in Fortran, C, C++, UPC, Java, Python.

TAU (Tuning and Analysis Utilities) is capable of gathering performance information through instrumentation of functions, methods, basic blocks, and statements as well as event-based sampling. All C++ language features are supported including templates and namespaces. The API also provides selection of profiling groups for organizing and controlling instrumentation. The instrumentation can be inserted in the source code using an automatic instrumentor tool based on the Program Database Toolkit (PDT), dynamically using DyninstAPI, at runtime in the Java Virtual Machine, or manually using the instrumentation API.

TAU's profile visualization tool, paraprof, provides graphical displays of all the performance analysis results, in aggregate and single node/context/thread forms. The user can quickly identify sources of performance bottlenecks in the application using the graphical interface.

In addition, TAU can generate event traces that can be displayed with the Vampir, Paraver or JumpShot trace visualization tools.

Version History

- 2015/03/31 TAU 2.24

How to Use

To use valgrind, load one of the valgrind modules as below

```
module load tau/2.24
```

For more details on using Valgrind, please read its online documents.

F.6.3. Valgrind

Valgrind is an instrumentation framework for building dynamic analysis tools. There are Valgrind tools that can automatically detect many memory management and threading bugs, and profile your programs in detail. You can also use Valgrind to build new tools.

Version History

- 2014/10/22 Valgrind 3.10.0
- 2013/11/08 Valgrind 3.9.0

How to Use

To use valgrind, load one of the valgrind modules as below

```
module load valgrind/3.9.0
```

or

```
module load valgrind/3.10.0
```

For more details on using Valgrind, please read its online documents.

F.7. Visualization

F.7.1. GNUPlot

GNUPlot is a portable command-line driven graphing utility for linux, OS/2, Windows, OSX, VMS, and many other platforms. It was originally created to allow scientists and students to visualize mathematical functions and data interactively, but has grown to support many non-interactive uses such as web scripting. Gnuplot supports many types of plots in either 2D and 3D. It can draw using lines, points, boxes, contours, vector fields, surfaces, and various associated text. It also supports various specialized plot types. Gnuplot supports many different types of output: interactive screen terminals (with mouse

and hotkey input), direct output to pen plotters or modern printers, and output to many file formats (eps, fig, jpeg, LaTeX, metafont, pbm, pdf, png, postscript, svg, ...). Gnuplot is easily extensible to include new output modes. Recent additions include an interactive terminal based on wxWidgets and the creation of mousable graphs for web display using the HTML5 canvas element.

Version History

- 2013/11/14 GNUPlot 4.6.4

How to Use

To use GNUPlot, load the module as below

```
module load gnuplot/4.6.4
```

Make sure to login the cluster or submit an interactively job with X-windows enabled. For more details on using GNUPlot, please read its online documents.

F.7.2. IDL

IDL is the trusted scientific programming language used across disciplines to extract meaningful visualizations from complex numerical data. With IDL you can interpret your data, expedite discoveries, and deliver powerful applications to market.

Version History

- 2014/01/15 IDL 8.3
- 2013/10/22 IDL 8.2 sp3

How to Use

To use IDL, load the module as below

```
module load idl/8.2_sp3
```

or

```
module load idl/8.3
```

Make sure to login the cluster or submit an interactively job with X-windows enabled. For more details on using IDL, please read its online documents.

F.7.3. matplotlib

matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB or Mathematica), web application servers, and six graphical user interface toolkits.

Version History

- 2015/01/28 matplotlib 1.4.2 based on Python 2.6
- 2014/06/20 matplotlib 1.3.1 based on Python 2.7
- 2013/11/11 matplotlib 1.3.1 based on Python 2.6

How to Use

To use matplotlib, load one of the modules as below

```
module load matplotlib/1.4.2
module load matplotlib/1.3.1
```

for Python 2.6 or the module

```
module load matplotlib/1.3.1_py27
```

for Python 2.7.

Make sure to login the cluster or submit an interactively job with X-windows enabled. For more details on using matplotlib, please read its online documents.

F.7.4. The NCAR Command Language (NCL)

The NCAR Command Language (NCL), a product of the Computational & Information Systems Laboratory at the National Center for Atmospheric Research (NCAR) and sponsored by the National Science Foundation, is a free interpreted language designed specifically for scientific data processing and visualization.

NCL has robust file input and output. It can read and write netCDF-3, netCDF-4 classic, netCDF-4, HDF4, binary, and ASCII data. It can read HDF-EOS2, HDF-EOS5, GRIB1, GRIB2, and OGR files (shapefiles, MapInfo, GMT, Tiger). It can be built as an OpenDAP client.

Version History

- 2013/11/06 NCL 6.1.2

How to Use

To use NCL, load the module as below

```
module load ncl/6.1.2
```

Make sure to login the cluster or submit an interactively job with X-windows enabled. For more details on using NCL, please read its online documents.

F.7.5. Ncview

Ncview is a visual browser for netCDF format files. Typically you would use ncview to get a quick and easy, push-button look at your netCDF files. You can view simple movies of the data, view along various dimensions, take a look at the actual data values, change color maps, invert the data, etc.

Version History

- 2015/02/25 Ncview 2.1.4

How to Use

To use Ncview, load the module as below

```
module load ncview/2.1.4
```

Make sure to login the cluster or submit an interactively job with X-windows enabled. For more details on using Ncview, please read its online documents.

F.7.6. OpenCV

OpenCV (Open Source Computer Vision) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 7 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

Version History

- 2014/04/04

How to Use

To use OpenCV, load the module as below

```
module load opencv/2.4.8
```

For more details on using OpenCV, please read its online documents.

F.8. Statistics and Mathematics Environments

F.8.1. R

R is language and environment for statistical computing and graphics. It provides a wide variety of statistical and graphical techniques (linear and nonlinear modeling, statistical tests, time series analysis, classification, clustering, ...). R is designed as a true computer language with control-flow constructions for iteration and alternation, and it allows users to add additional functionality by defining new functions. For computationally-intensive tasks, C, C++, and Fortran code can be linked and called at run time. Advanced users can write C or Java code to manipulate R objects directly.

Version History

- 2014/08/26 R 3.1.1
- 2013/11/01 R 3.0.2

How to Use

To use R, load one of the R modules as below

```
module load R/3.1.1
module load R/3.0.2
```

Currently loaded libraries:

```
abind                   Combine multi-dimensional arrays
base                    The R Base Package
boot                    Bootstrap Functions (originally by Angelo Canty
                        for S)
class                   Functions for Classification
clue                    Cluster ensembles
cluster                 Cluster Analysis Extended Rousseeuw et al.
cmm                     Categorical Marginal Models
coda                    Output analysis and diagnostics for MCMC
codetools               Code Analysis Tools for R
compiler                The R Compiler Package
cubature                Adaptive multivariate integration over
                        hypercubes
datasets                The R Datasets Package
degreenet               Models for Skewed Count Distributions Relevant
                        to Networks
drm                     Regression and association models for repeated
                        categorical data
e1071                   Misc Functions of the Department of Statistics
                        (e1071), TU Wien
ergm                    Fit, Simulate and Diagnose Exponential-Family
                        Models for Networks
flexclust               Flexible Cluster Algorithms
foreign                 Read Data Stored by Minitab, S, SAS, SPSS,
                        Stata, Systat, dBase, ...
gamlss                  Generalised Additive Models for Location Scale
                        and Shape.
gamlss.data             GAMLSS Data.
gamlss.dist             Distributions to be used for GAMLSS modelling.
gamlss.mx               A GAMLSS add on package for fitting mixture
                        distributions
gee                     Generalized Estimation Equation solver
geepack                 Generalized Estimating Equation Package
graphics                The R Graphics Package
grDevices               The R Graphics Devices and Support for Colours
                        and Fonts
```

```
grid                    The Grid Graphics Package
gtools                  Various R programming tools
KernSmooth              Functions for kernel smoothing for Wand & Jones
                        (1995)
ks                      Kernel smoothing
latentnet               Latent position and cluster models for
                        statistical networks
lattice                 Lattice Graphics
lme4                    Linear mixed-effects models using Eigen and S4
magic                   create and investigate magic squares
maps                    Draw Geographical Maps
MASS                    Support Functions and Datasets for Venables and
                        Ripley's MASS
Matrix                  Sparse and Dense Matrix Classes and Methods
methods                 Formal Methods and Classes
mgcv                    Mixed GAM Computation Vehicle with GCV/AIC/REML
                        smoothness estimation
minpack.lm              R interface to the Levenberg-Marquardt
                        nonlinear least-squares algorithm found in
                        MINPACK, plus support for bounds
minqa                   Derivative-free optimization algorithms by
                        quadratic approximation
misc3d                  Miscellaneous 3D Plots
mixtools                Tools for analyzing finite mixture models
modeltools              Tools and Classes for Statistical Models
moments                 Moments, cumulants, skewness, kurtosis and
                        related tests
multicool               Permutations of multisets in cool-lex order.
mvtnorm                 Multivariate Normal and t Distributions
network                 Classes for Relational Data
nlme                    Linear and Nonlinear Mixed Effects Models
nnet                    Feed-forward Neural Networks and Multinomial
                        Log-Linear Models
np                      Nonparametric kernel smoothing methods for
                        mixed data types
numDeriv                Accurate Numerical Derivatives
orth                    Multivariate Logistic Regressions Using
                        Orthogonalized Residuals.
orthpoly                orthonormal polynomials
parallel                Support for Parallel computation in R
plyr                    Tools for splitting, applying and combining
                        data
pracma                  Practical Numerical Math Functions
Rcpp                    Seamless R and C++ Integration
RcppEigen               Rcpp integration for the Eigen templated linear
                        algebra library.
rgl                     3D visualization device system (OpenGL)
rlecuyer                R interface to RNG with multiple streams
R.methodsS3             Utility function for defining S3 methods
robustbase              Basic Robust Statistics
R.oo                    R object-oriented programming with or without
                        references
rpart                   Recursive Partitioning
scatterplot3d           3D Scatter Plot
segmented               Segmented relationships in regression models
                        with breakpoints/changepoints estimation
shapes                  Statistical shape analysis
sna                     Tools for Social Network Analysis
snow                    Simple Network of Workstations
snowFT                  Fault Tolerant Simple Network of Workstations
spatial                 Functions for Kriging and Point Pattern
                        Analysis
splines                 Regression Spline Functions and Classes
statnet.common          Common R Scripts and Utilities Used by the
                        Statnet Project Software
stats                   The R Stats Package
stats4                  Statistical Functions using S4 Classes
survival                Survival Analysis
tcltk                   Tcl/Tk Interface
tensor                  Tensor product of arrays
tools                   Tools for Package Development
trust                   Trust Region Optimization
utils                   The R Utils Package
```

```
zoo                     S3 Infrastructure for Regular and Irregular
                        Time Series (Z's ordered observations)
```

Contact admin if you need to install addition libraries in R.

Job Script Example

An example job script on running R is given as below and located at

```
#!/bin/sh
#PBS -N job
#PBS -l nodes=1:ppn=1
#PBS -l cput=10:00:00
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe

#============================================================#
# USER CONFIG
#============================================================#
INPUT_FILE="lapack.R"
OUTPUT_FILE="output.log"
MODULE_NAME="R/3.1.1"
PROGRAM_NAME="Rscript"
#============================================================#
# MODULE is loaded
#============================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE
#============================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#============================================================#
cd $PBS_O_WORKDIR
# Run a system wide sequential application
$PROGRAM_NAME $INPUT_FILE >& $OUTPUT_FILE

#============================================================#
# ALL DONE
#============================================================#
```

The example test case can be found at `/hpc/tmp/examples/R/3.1.1` For more details on using R, please refer to R online documents.

F.9. Computational Physics and Chemistry

F.9.1. ABINIT

ABINIT is a package whose main program allows one to find the total energy, charge density and electronic structure of systems made of electrons and nuclei (molecules and periodic solids) within Density Functional Theory (DFT), using pseudopotentials and a planewave basis. ABINIT also includes options to optimize the geometry according to the DFT forces and stresses, or to perform molecular dynamics simulations using these forces, or to generate dynamical matrices, Born effective charges, and dielectric tensors. Excited states can be computed within the Time-Dependent Density Functional Theory (for molecules), or within Many-Body Perturbation Theory (the GW approximation). In addition to the main ABINIT code, different utility programs are provided.

Version History

- 2015/02/05 ABINIT 7.10.2 Intel Build
- 2014/11/07 ABINIT 7.8.2 Intel Build

How to Use

To use ABINIT, load one of the following modules

```
module load abinit/7.10.2_itl
```

or

```
module load abinit/7.8.2_itl
```

Job Script Example

An example job script is similar to the following `jobscript` file:

```
#!/bin/sh
#PBS -l cput=30:00:00
#PBS -l pvmem=1GB
#PBS -j eo
#PBS -e test.err
#PBS -m abe
#PBS -l nodes=1:ppn=4

#=======================================================#
# USER CONFIG
#=======================================================#
INPUT_FILE=tH1.file
OUTPUT_FILE=output.log
MODULE_NAME=abinit/7.8.2_itl
PROGRAM_NAME=abinit
COPY_SCRATCH_BACK=true
#=======================================================#
# MODULE is loaded
#=======================================================#
source /etc/profile.d/modules.sh
module load $MODULE_NAME
NP=`wc -l < $PBS_NODEFILE`
cat $PBS_NODEFILE

#=======================================================#
# SCRATCH directory is created at the local disks
#=======================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
fi

#=======================================================#
# TRANSFER input files to the scratch directory
#=======================================================#
# just copy input file
cp $PBS_O_WORKDIR/$INPUT_FILE $SCRDIR
readarray FILE_TO_COPY < $PBS_O_WORKDIR/$INPUT_FILE
echo "COPY FILES FROM THE WORKING DIRECTORY TO THE SCRATCH DIRECTORY:"
for file in ${FILE_TO_COPY[@]}
do
    if [ -f "$PBS_O_WORKDIR/$file" ]; then
        echo "COPY FILE " $PBS_O_WORKDIR/$file " TO " $SCRDIR
        cp $PBS_O_WORKDIR/$file $SCRDIR
    fi
done
# copy everything (Option)
```

```
#cp $PBS_O_WORKDIR/* $SCRDIR

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $SCRDIR
mpirun -np $NP $PROGRAM_NAME < $INPUT_FILE > $PBS_O_WORKDIR/$OUTPUT_FILE

#=========================================================#
# RESULTS are migrated back to the working directory
#=========================================================#
if [[ "$COPY_SCRATCH_BACK" == *true* ]]
then
    echo "COPY RESULTS BACK TO THE WORKING DIRECTORY:" $PBS_O_WORKDIR
    for file in ${FILE_TO_COPY[@]}
    do
        if [ -f "$SCRDIR/$file" ]; then
            cp -rp $SCRDIR/$file $PBS_O_WORKDIR/$file
        fi
    done
fi

#=========================================================#
# DELETING the local scratch directory
#=========================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi

#=========================================================#
# ALL DONE
#=========================================================#
```

In the above example job script, the Input files are copied to the local scratch directory for running the simulation. The results are copied back to the working directory after the job terminates. Users just need to revise the contents within the 'USER CONFIG' portion to specify the input and output files. The example file can be found at

```
/hpc/tmp/examples/abinit
```

For more details on using Abinit, please read its online documents.

F.9.2. Atomic Simulation Environment (ASE)

The Atomistic Simulation Environment (ASE) is a set of tools and Python modules for setting up, manipulating, running, visualizing and analyzing atomistic simulations. The code is freely available under the GNU LGPL license.

Version History

- 2015/01/28 ASE 3.8.1

How to Use

To use ASE, adding

```
module load ase/3.8.1
```

in your job script. For more details on using ASE, please read its online documents.

F.9.3. Atomistix ToolKit (ATK)

Atomistix ToolKit (ATK) is a software package that offers unique capabilities for simulating nanostructures on the atomic scale

- NEGF simulations to study transport properties like I-V characteristics of nanoelectronic devices
- Powerful combination of DFT, semi-empirical tight-binding, classical potentials in the same package
- Advanced graphical user interface for building complicated structures like interfaces and transport systems
- Plugin-based platform which can interface with external codes
- Python scripting interface

Version History

- 2013/03/10 ATK 13.8.1

How to Use

ATK is a commercial licensed software. Please contact HPC admin on how to access it.
For more details on using ATK, please read its online documents.

F.9.4. AutoDock and AutoDock Vina

AutoDock is a suite of automated docking tools. It is designed to predict how small molecules, such as substrates or drug candidates, bind to a receptor of known 3D structure. Current distributions of AutoDock consist of two generations of software: AutoDock 4 and AutoDock Vina. AutoDock 4 actually consists of two main programs: autodock performs the docking of the ligand to a set of grids describing the target protein; autogrid pre-calculates these grids.

Version History

- 2013/11/11 AutoDockSuite 4.2.5.1
- 2013/11/11 AutoDockVina 1.1.2

How to Use

To use AutoDock, adding

```
module load autodocksuite/4.2.5.1
```

in your job script.
To use AutoDock Vina, adding

```
module load autodock_vina/1.1.2_bin
```

in your job script.

You need to specify the requested core number in the '`--cpu`' flag, i.e. '`--cpu 1`' for a single core job or '`--cpu N`' in the job requesting 'N' cores.

For more details on using AutoDock Vina, please read its online manual.

F.9.5. CP2K

CP2K is a program to perform atomistic and molecular simulations of solid state, liquid, molecular, and biological systems. It provides a general framework for different methods such as e.g., density functional theory (DFT) using a mixed Gaussian and plane waves approach (GPW) and classical pair and many-body potentials.

Version History

- 2015/01/13 CP2K 2.6.0 GCC Build
- 2014/11/21 CP2K 2.5.1 GCC Build
- 2013/11/12 CP2K 2.4.0 GCC Build

How to Use

To use CPMD, load one of the following modules

```
module load cp2k/2.4_gcc_acml
```

or

```
module load cp2k/2.5.1_gcc_acml
```

or

```
module load cp2k/2.6.0_gcc_acml
```

Job Script Example

An example job script on using CP2K 2.6.0 is given as below:

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l pvmem=4GB
#PBS -l cput=100:00:00
#PBS -j eo
#PBS -e test.err
#PBS -N test
#PBS -m ae

#=====================================================#
# USER CONFIG
#=====================================================#
INPUT_FILE=H2O-64.inp
OUTPUT_FILE=H2O-64.out
MODULE_NAME=cp2k/2.6.0_gcc_acml
PROGRAM_NAME=cp2k.popt
# Set as true if you need those scratch files.
COPY_SCRATCH_BACK=true
```

```
#=========================================================#
# MODULE is loaded
#=========================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=========================================================#
# SCRATCH directory is created at the local disks
#=========================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
fi

#=========================================================#
# TRANSFER input files to the scratch directory
#=========================================================#
# just copy input file
# cp $PBS_O_WORKDIR/$INPUT_FILE $SCRDIR
# copy everything (Option)
cp $PBS_O_WORKDIR/* $SCRDIR

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $SCRDIR
# Run a MPI application
mpirun -np $NP $PROGRAM_NAME $INPUT_FILE >& $PBS_O_WORKDIR/$OUTPUT_FILE

#=========================================================#
# RESULTS are migrated back to the working directory
#=========================================================#
if [[ "$COPY_SCRATCH_BACK" == *true* ]]
then
    echo "COPYING SCRACH FILES TO " $PBS_O_WORKDIR
    cp -rp $SCRDIR/* $PBS_O_WORKDIR
    if [ $? != 0 ]; then
        {
            echo "Sync ERROR: problem copying files from $tdir to $PBS_O_WORKDIR;"
            echo "Contact HPC admin for a solution."
            exit 1
        }
    fi
fi

#=========================================================#
# DELETING the local scratch directory
#=========================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi
#=========================================================#
# ALL DONE
#=========================================================#
```

The example file can be found at

```
/hpc/tmp/examples/cp2k
```

NOTE: You must specify the absolute path of the potential files in the input files.

For more details on using CP2K, please read its online documents.

F.9.6. CPMD

The CPMD code is a plane wave/pseudopotential implementation of Density Functional Theory, particularly designed for ab-initio molecular dynamics. Its first version was developed by Jurg Hutter at IBM Zurich Research Laboratory starting from the original Car-Parrinello codes. During the years many people from diverse organizations contributed to the development of the code and of its pseudopotential library:contains series of electronic structure programs, used by chemists, chemical engineers, biochemists, physicists and other scientists worldwide.

Version History

- 2013/11/20 CPMD 1.17.2 PGI build

How to Use

To use CPMD, load modules as below

```
module load cpmd/3.17.1_pgi
```

Job Script Example

An example job script on using cpmd-3.17.1 is given as below:

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l cput=30:00:00,pvmem=800MB
#PBS -j eo
#PBS -e test.err
#PBS -N cpmd.job

#=========================================================#
# USER CONFIG
#=========================================================#
INPUT_FILE=si64.inp
OUTPUT_FILE=si64.out
MODULE_NAME=cpmd/3.17.1_pgi
PROGRAM_NAME=cpmd.x
PP_LIBRARY_PATH=/hpc/software/package/cpmd/3.17.1_pgi/pseudo_extlib
# Set as true if you need those scratch files.
COPY_SCRATCH_BACK=true

#=========================================================#
# MODULE is loaded
#=========================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE
export PP_LIBRARY_PATH=$PP_LIBRARY_PATH

#=========================================================#
# SCRATCH directory is created at the local disks
#=========================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
fi

#=========================================================#
# TRANSFER input files to the scratch directory
#=========================================================#
```

```
# just copy input file
cp $PBS_O_WORKDIR/$INPUT_FILE $SCRDIR
# copy everything (Option)
#cp $PBS_O_WORKDIR/* $SCRDIR

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
echo "START TO RUN WORK"
cd $SCRDIR
# Run a MPI application
mpirun -np $NP $PROGRAM_NAME $INPUT_FILE >& $PBS_O_WORKDIR/$OUTPUT_FILE

#=========================================================#
# RESULTS are migrated back to the working directory
#=========================================================#
if [[ "$COPY_SCRATCH_BACK" == *true* ]]
then
    echo "COPYING SCRACH FILES TO " $PBS_O_WORKDIR/$PBS_JOBID
    cp -rp $SCRDIR/* $PBS_O_WORKDIR
    if [ $? != 0 ]; then
        {
            echo "Sync ERROR: problem copying files from $tdir to $PBS_O_WORKDIR;"
            echo "Contact HPC admin for a solution."
            exit 1
        }
    fi
fi

#=========================================================#
# DELETING the local scratch directory
#=========================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi
#=========================================================#
# ALL DONE
#=========================================================#
```

The example file can be found at

```
/hpc/tmp/examples/cpmd/3.17.1
```

NOTE: Must specify the `PP_LIBRARY_PATH` variable in the `USER CONFIG` portion with the directory containing the pseudopotential files in use. The default pseudopotential directory is `/hpc/software/package/cpmd/3.17.1_pgi/pseudo_extlib`.

For more details on using CPMD, please read its online documents.

F.9.7. DOCK

DOCK addresses the problem of "docking" molecules to each other. In general, "docking" is the identification of the low-energy binding modes of a small molecule, or ligand, within the active site of a macromolecule, or receptor, whose structure is known. A compound that interacts strongly with, or binds, a receptor associated with a disease may inhibit its function and thus act as a drug. Solving the docking problem computationally requires an accurate representation of the molecular energetics as well as an efficient algorithm to search the potential binding modes.

Version History

- 2013/11/12 DOCK 6.5 GCC Build

How to Use

To use DOCK, load modules as below

```
module load dock/6.5_mpi
```

Job Script Example

```
#!/bin/bash
#PBS -l cput=0:30:00
#PBS -l pvmem=400MB
#PBS -N dock.job
#PBS -l nodes=1:ppn=4
#PBS -m abe
#PBS -j eo
#PBS -e test.err

#=======================================================#
# USER CONFIG
#=======================================================#
INPUT_FILE=mpi.dockin
OUTPUT_FILE=mpi.dockout
MODULE_NAME=dock/6.5_mpi
PROGRAM_NAME=dock6.mpi
# Set as true if you need those scratch files.
COPY_SCRATCH_BACK=true
#=======================================================#
# MODULE is loaded
#=======================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=======================================================#
# SCRATCH directory is created at the local disks
#=======================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
fi

#=======================================================#
# TRANSFER input files to the scratch directory
#=======================================================#
# just copy input file
cp $PBS_O_WORKDIR/$INPUT_FILE $SCRDIR
# copy everything (Option)
#cp $PBS_O_WORKDIR/* $SCRDIR

#=======================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=======================================================#
cd $SCRDIR
# Run a MPI application (Option)
 mpirun -np $NP $PROGRAM_NAME -i $INPUT_FILE -o $PBS_O_WORKDIR/$OUTPUT_FILE

#=======================================================#
# RESULTS are migrated back to the working directory
#=======================================================#
if [[ "$COPY_SCRATCH_BACK" == *true* ]]
then
```

```
    echo "COPYING SCRACH FILES TO " $PBS_O_WORKDIR/$PBS_JOBID
    cp -rp $SCRDIR/* $PBS_O_WORKDIR
    if [ $? != 0 ]; then
        {
            echo "Sync ERROR: problem copying files from $tdir to $PBS_O_WORKDIR;"
            echo "Contact HPC admin for a solution."
            exit 1
        }
    fi
fi

#=========================================================#
# DELETING the local scratch directory
#=========================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi
#=========================================================#
# ALL DONE
#=========================================================#
```

The example test can be found at `/hpc/tmp/examples/dock/6.5` NOTE: Specify the necessary auxiliary files with their absolute path in the input file. Such as

```
ligand_atom_file            /hpc/tmp/examples/dock/6.5/multi_lig.mol2
receptor_site_file          /hpc/tmp/examples/dock/6.5/struct.sph
grid_score_grid_prefix      /hpc/tmp/examples/dock/6.5/grid_generation/grid
vdw_defn_file               /hpc/tmp/examples/dock/6.5/parameters/vdw_AMBER_parm99.defn
flex_defn_file              /hpc/tmp/examples/dock/6.5/parameters/flex.defn
flex_drive_file             /hpc/tmp/examples/dock/6.5/parameters/flex_drive.tbl
```

F.9.8. GAMESS

GAMESS is a program for ab initio molecular quantum chemistry. Briefly, GAMESS can compute SCF wavefunctions ranging from RHF, ROHF, UHF, GVB, and MCSCF. Correlation corrections to these SCF wavefunctions include Configuration Interaction, second order perturbation Theory, and Coupled-Cluster approaches, as well as the Density Functional Theory approximation.

Please make registration on GAMESS website before accessing to GAMESS package deployed at the cluster.

Version History

- 2013/10/24 GAMESS May_2013_R1

How to Use

To use GAMESS, adding

```
module load gamess/May_2013_R1
```

in your job script.

Job Script Example

An example job of running GAMESS calculations can be found at `/hpc/tmp/examples/gamess/May_2013_R1` and the job script is shown as below:

```
#!/bin/sh
#PBS -N gamess.job
#PBS -l cput=100:00:00
#PBS -l pvmem=2GB
#PBS -j eo
#PBS -e test.err
#PBS -l nodes=1:ppn=8

#=======================================================#
# USER CONFIG
#=======================================================#
INPUT_FILE=exam01.inp
OUTPUT_FILE=exam01.out
MODULE_NAME=gamess/May_2013_R1
PROGRAM_NAME=rungms
#=======================================================#
# MODULE is loaded
#=======================================================#
NP=`wc -l < $PBS_NODEFILE`
VER=gft
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=======================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=======================================================#
cd $PBS_O_WORKDIR
rungms $INPUT_FILE $VER $NP >& $OUTPUT_FILE
#=======================================================#
# DELETING the local scratch directory
#=======================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi
#=======================================================#
# ALL DONE
#=======================================================#
```

An example case can be found at `/hpc/tmp/examples/gamess/May_2013_R1`.

For more details on using GAMESS, please read its online documents.

F.9.9. GATE

GATE (Geant4 Application for Tomographic Emission) is an advanced open source software developed by the international OpenGATE collaboration and dedicated to numerical simulations in medical imaging and radiotherapy. It currently supports simulations of Emission Tomography (Positron Emission Tomography - PET and Single Photon Emission Computed Tomography - SPECT), Computed Tomography (CT) and Radiotherapy experiments. Using an easy-to-learn macro mechanism to configurate simple or highly sophisticated experimental settings, GATE now plays a key role in the design of new medical imaging devices, in the optimization of acquisition protocols and in the development and assessment of image reconstruction algorithms and correction techniques. It can also be used for dose calculation in radiotherapy experiments.

Version History

- 2014/05/14 GATE 7.0
- 2014/04/24 GATE 6.2

How to Use

To use GATE, load one of the following modules in your job script.

```
module load gate/6.2
```

or

```
module load gate/7.0
```

GATE 6.2 is built on Geant 4.9.5p02 and GATE 7.0 is build on Geant 4.9.6p03.

Batch Mode

An example job of running GATE calculations can be found at `/hpc/tmp/examples/gate/6.2/benchOET/batch` and the job script is shown as below:

```
#!/bin/sh
#PBS -N job
#PBS -l nodes=1:ppn=1
#PBS -l cput=1000:00:00
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe

#=====================================================#
# USER CONFIG
#=====================================================#
INPUT_FILE="benchPET.mac"
OUTPUT_FILE="bench.out"
MODULE_NAME="gate/6.2"
PROGRAM_NAME="Gate"
#=====================================================#
# MODULE is loaded
#=====================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=====================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=====================================================#
cd $PBS_O_WORKDIR
# Run a system wide sequential application
$PROGRAM_NAME < $INPUT_FILE >& $OUTPUT_FILE

#=====================================================#
# ALL DONE
#=====================================================#
```

Please note, '`Gate`' is not a parallelized program. You can only request single core to run it in the batch mode.

Cluster Mode

To reduce the overall computing time of 'GATE' experiments, you could split a single batch simulation into several parts and run them simultaneously. This will significantly shortens the setup time and provides fast data output handling. An example job under
/hpc/tmp/examples/gate/v6.2/benchPET/cluster
is used to illustrate how to run Gate in the cluster mode.

- Enter the working directory and prepare your input file.

```
-sh-4.1$ cd /hpc/tmp/examples/gate/6.2/benchPET/cluster
-sh-4.1$ ls
camera.mac     GateMaterials.db  physics.mac  visu.mac
benchPET.mac   digitizer.mac     phantom.mac  sources.mac
```

- Load the 'gate' module in the command line.

```
-sh-4.1$ module load gate/6.2
```

- Subdivide the simulation macro into fully resolved, non-parameterized macros as below:

```
-sh-4.1$ rungjs -numberofsplits 10 benchPET.mac
```

  Here '10' means the number of sub-macro and 'benchPET.mac' is the name of the batch simulation macro. After that, you will get a hidden directory '.Gate' under the current working directory. It has a subdirectory called as the macro name, that contains the following files:

```
-rw-r--r-- 1 ruiy guru  9256 Apr 30 16:44 benchPET1.mac
-rw-r--r-- 1 ruiy guru  1898 Apr 30 16:44 benchPET1.pbs
-rw-r--r-- 1 ruiy guru  9256 Apr 30 16:44 benchPET2.mac
-rw-r--r-- 1 ruiy guru  1898 Apr 30 16:44 benchPET2.pbs
...
-rw-r--r-- 1 ruiy guru  1439 Apr 30 16:44 benchPET.split
```

- A script suffixed with '.submit', i.e. 'benchPET.submit' in the example directory, will also be created from the above step. Run this script to submit subdivision jobs.

```
-sh-4.1$ ./benchPET.submit
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET1.pbs
209389.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET2.pbs
209390.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET3.pbs
209391.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET4.pbs
209392.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET5.pbs
209393.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET6.pbs
209394.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET7.pbs
209395.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET8.pbs
209396.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET9.pbs
209397.hpc.local
qsub /hpc/tmp/examples/gate/v6.2/benchPET/cluster/.Gate/benchPET/benchPET10.pbs
209398.hpc.local
```

  These jobs will produce '.root' files as the results.

- After all subdivision jobs complete, copy the file '.Gate/benchPET/benchPET.split' to the current working directory and run 'rungjm' command to merge the output files into a single file (make sure the 'gate' module is loaded):

```
-sh-4.1$ pwd
/hpc/tmp/examples/gate/6.2/benchPET/cluster
-sh-4.1$ cp .Gate/benchPET/benchPET.split .
-sh-4.1$ rungjm ./benchPET.split
Combining: benchmarkPET1.root benchmarkPET2.root benchmarkPET3.root
benchmarkPET4.root benchmarkPET5.root benchmarkPET6.root benchmarkPET7.root
benchmarkPET8.root benchmarkPET9.root benchmarkPET10.root -> benchmarkPET.root
```

The final output file '`benchmarkPET.root`' is created.

For more details on using GATE, please read its online documents.

F.9.10. GAUSSIAN

GAUSSIAN contains series of electronic structure programs, used by chemists, chemical engineers, biochemists, physicists and other scientists worldwide. Starting from the fundamental laws of quantum mechanics, GAUSSIAN 09 predicts the energies, molecular structures, vibrational frequencies and molecular properties of molecules and reactions in a wide variety of chemical environments.

GAUSSIAN is commercial licensed software. Please contact admin for further information.

Version History

- 2013/11/20 Gaussian G09A02
- 2013/10/22 Gaussian G09C01

How to Use

To use Gaussian, load modules as below

```
module load gaussian/g09a02
```

or

```
module load gaussian/g09c01
```

for different versions.

Job Script Example

```
#!/bin/bash
#PBS -l nodes=1:ppn=8
#PBS -l pmem=4GB
#PBS -l cput=100:00:00
#PBS -j eo
#PBS -e job.std.out
#PBS -N g09c01.job
#PBS -m ae

#=====================================================#
# USER CONFIG
#=====================================================#
INPUT_FILE=test397.com
OUTPUT_FILE=test397.log
MODULE_NAME=gaussian/g09c01
PROGRAM_NAME=g09

#=====================================================#
```

```
# MODULE is loaded
#=========================================================#
NP='wc -l < $PBS_NODEFILE'
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $PBS_O_WORKDIR
$PROGRAM_NAME < $INPUT_FILE >& $OUTPUT_FILE

#=========================================================#
# ALL DONE
#=========================================================#
```

The example job can be found at `/hpc/tmp/examples/gaussian/g09c01`.

NOTE: Gaussian is parallelized using OpenMP which can only utilize CPUs within a single node. Always request CPUs within one node by using `#PBS -l nodes=1:ppn=`N with $1 \leq N \leq 32$.

For more details on using Gaussian, please read its online documents.

F.9.11. Geant

Geant is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The two main reference papers for Geant4 are published in Nuclear Instruments and Methods in Physics Research A 506 (2003) 250-303, and IEEE Transactions on Nuclear Science 53 No. 1 (2006) 270-278.

Version History

- 2015/02/26 Geant 4.9.6p04
- 2014/12/11 Geant 4.10.01
- 2014/11/04 Geant 4.10.0p03
- 2014/08/11 Geant 4.10.0p02
- 2014/05/14 Geant 4.10.0p01
- 2014/05/14 Geant 4.9.6p03
- 2013/12/14 Geant 4.10.00
- 2013/11/11 Geant 4.9.6p01
- 2013/10/23 Geant 4.9.6p02
- 2013/10/23 Geant 4.9.5p02

How to Use

To use Geant, load one of the following modules in your job script.

```
module load geant/4.10.1
module load geant/4.10.0p03
module load geant/4.10.0p02
module load geant/4.10.0p01
```

```
module load geant/4.10.0
module load geant/4.9.5p02
module load geant/4.9.6p01
module load geant/4.9.6p02
module load geant/4.9.6p03
module load geant/4.9.6p04
```

This will add the Geant 4 binary path to the environment variable `PATH` and the library path `LD_LIBRARY_PATH`. All Geant data files will be set with the appropriate variables.

For more details on using Geant4, please read its online documents.

F.9.12. GPAW

GPAW is a density-functional theory (DFT) Python code based on the projector-augmented wave (PAW) method and the atomic simulation environment (ASE). It uses real-space uniform grids and multigrid methods, atom-centered basis-functions or plane-waves. Read more about the Algorithms used.
Feature List: Plane-waves,Finite-difference, LCAO, XC-functionals, DFT+U, GLLB-SC, DOS, STM, Wannier functions, delta-SCF, XAS, Jellium, TDDFT, LRTDDFT (molecules), LRTDDFT (extended systems), Transport, NEGF-transport, Keldysh, GF-transport ..., RPA-correlation, GW, BSE, Parallelization.

Version History

- 2015/01/28 GPAW 0.10.0

How to Use

To use GPAW, load module as below

```
module load gpaw/0.10.0
```

Job Script Example

An example job scrpit on submitting parallel GPAW jobs to the PBS is given below:

```
#!/bin/sh
#PBS -N job
#PBS -l nodes=1:ppn=4
#PBS -l cput=10:00:00
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe

#=======================================================#
# USER CONFIG
#=======================================================#
INPUT_FILE="bulk.py"
OUTPUT_FILE="output.log"
MODULE_NAME="gpaw/0.10.0"
PROGRAM_NAME="gpaw-python"

#=======================================================#
# MODULE is loaded
#=======================================================#
```

```
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $PBS_O_WORKDIR
# Run the MPI program.
mpirun -np $NP $PROGRAM_NAME $INPUT_FILE >& $OUTPUT_FILE

#=========================================================#
# ALL DONE
#=========================================================#
```

The example job can be found at `/hpc/tmp/examples/gpaw/0.10.0`.
For more details on using GROMACS, please read its manual.

F.9.13. GROMACS

GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the nonbonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.

Version History

- 2013/12/19 GROMACS 4.6.5 GCC OpenMPI build

How to Use

To use GROMACS, load module as below

```
module load gromacs/4.6.5_gcc_openmpi
```

Job Script Example

An example job scrpit on submitting parallel GROMACS jobs to the PBS is given below:

```
#!/bin/bash
#PBS -l walltime=01:00:10,vmem=800MB
#PBS -l nodes=1:ppn=4
#PBS -j eo
#PBS -e test.err
#PBS -m abe
#PBS -N gromacs.job

#=========================================================#
# USER CONFIG
#=========================================================#
# Module name
MODULE_NAME=gromacs/4.6.5_gcc_openmpi

#=========================================================#
# MODULE is loaded
```

```
#=============================================================#
NP='wc -l < $PBS_NODEFILE'
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=============================================================#
# SCRATCH directory is created at the local disks
#=============================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
fi
export TMPDIR=$SCRDIR
export TMP_DIR=$SCRDIR

#=============================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=============================================================#
cd $PBS_O_WORKDIR
grompp_mpi_d -f min_stpst.mdp -c methane.gro -o min_stpst.tpr -p methane.top
mpirun -np $NP mdrun_mpi_d -s min_stpst.tpr -c methane.gro -g min_stpst.log -e min_stpst.ene


#=============================================================#
# DELETING the local scratch directory
#=============================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi

#=============================================================#
# ALL DONE
#=============================================================#
```

The GROMACS executables are available with the suffix '_d' as it is a double precision version. Please note the extra suffix '_mpi' for the MPI-aware utility `mdrun_mpi_d`.

The example job can be found at `/hpc/tmp/examples/gromacs/4.6.5_openmpi`.

For more details on using GROMACS, please read its online tutorials.

F.9.14. MGLTools

MGLTools is a software developed at the Molecular Graphics Laboratory (MGL) of the Scripps Research Institute for visualization and analysis of molecular structures. It has three main applications: AutoDockTools is graphical front-end for setting up and running AutoDock - an automated docking software designed to predict how small molecules, such as substrates or drug candidates, bind to a receptor of known 3D structure; PMV is a powerful molecular viewer that has a number of customizable features and comes with many pluggable commands ranging from displaying molecular surfaces to advanced volume rendering; and Vision is a visual-programming environment in which a user can interactively build networks describing novel combinations of computational methods, and yielding new visualizations of their data without actually writing code.

Version History

- 2013/11/11 MGLTools 1.5.6

How to Use

To use MGLTools, first make sure your local desktop has X-Server support. After logging on the cluster with X forwarding, submit an interactive job with X support as below:

```
qsub -I -X -q short
```

Then load the MGLTools module:

```
module load mgltools/1.5.6
```

Type the command to run different applications such as `adt`, `pmv` and `vision`.

After finishing all work, type `exit` to quit the interactive job.

For more details on using MGLTools, please read its online tutorials.

F.9.15. MOLDEN

Molden is a package for displaying Molecular Density from the *Ab Initio* packages such as GAMESS-UK, GAMESS-US and GAUSSIAN and the Semi-Empirical packages such as Mopac/Ampac, it also supports a number of other programs via the Molden Format. Molden reads all the required information from the GAMESS/GAUSSIAN outputfile. Molden is capable of displaying Molecular Orbitals, the electron density and the Molecular minus Atomic density. Either the spherically averaged atomic density or the oriented ground state atomic density can be subtracted for a number of standard basis sets. Molden supports contour plots, 3-d grid plots with hidden lines and a combination of both. It can write a variety of graphics instructions; postscript, XWindows, VRML, povray, OpenGL, tekronix4014, hpgl, hp2392 and Figure. Both Xwindows and OpenGL versions of Molden are also capable of importing and displaying of chemx, PDB, and a variety of mopac/ampac files and lots of other formats. Molden also can animate reaction paths and molecular vibrations. It can calculate and display the true or Multipole Derived Electrostatic Potential and atomic charges can be fitted to the Electrostatic Potential calculated on a Connolly surface. Molden also features an stand alone forcefield program ambfor, which can optimise geometries with the combined Amber (protein) and GAFF (small molecules) force fields. Atom typing can be done automatically and interactively from within Molden, as well as firing optimisation jobs. Molden has a powerful Z-matrix editor which give full control over the geometry and allows you to build molecules from scratch, including polypeptides. Molden was also submitted to the QCPE (dead?) (QCPE619), allthough the Xwindows version is considerably running behind on the current one.

Version History

- 2014/04/07 Molden 5.1

How to Use

To use Molden, please submit an interactive job to 'short' queue and then load the module 'molden', i.e.

```
-sh-4.1$ qsub -I -X -q short
-sh-4.1$ module load molden/5.1.0
-sh-4.1$ xcrysden
```

Remember typing ‘exit’ after completing all works to quit the interactive job.

F.9.16. NAMD

NAMD is a parallel, object-oriented molecular dynamics code designed for high-performance simulation of large biomolecular systems.

Version History

- 2015/01/06 NAMD 2.10
- 2013/10/22 NAMD 2.9

How to Use

To use NAMD, load one of the modules as below

```
module load namd/2.10
module load namd/2.9_bin
```

Job Script Example

```
#!/bin/bash
#PBS -l nodes=1:ppn=8
#PBS -l cput=30:00:00
#PBS -j eo
#PBS -e test.err
#PBS -N namd.job

#=====================================================#
# USER CONFIG
#=====================================================#
# Input file name
INPUT_FILE=apoa1.namd
# Output file name
OUTPUT_FILE=test.log
# Module name
MODULE_NAME=namd/2.10
# Program name
PROGRAM_NAME=namdrun

#=====================================================#
# MODULE is loaded
#=====================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=====================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=====================================================#
cd $PBS_O_WORKDIR
$PROGRAM_NAME  $INPUT_FILE >& $OUTPUT_FILE

#=====================================================#
# ALL DONE
#=====================================================#
```

The example test can be found at `/hpc/tmp/examples/namd/2.10`

For more details on using NAMD, please read its online tutorials.

F.9.17. NWChem

NWChem aims to provide its users with computational chemistry tools that are scalable both in their ability to treat large scientific computational chemistry problems efficiently, and in their use of available parallel computing resources from high-performance parallel supercomputers to conventional workstation clusters. NWChem software can handle

- Biomolecules, nanostructures, and solid-state
- From quantum to classical, and all combinations
- Gaussian basis functions or plane-waves
- Scaling from one to thousands of processors
- Properties and relativity

Version History

- 2014/05/11 nwchem 6.3 R2 Intel build

How to Use

To use NWChem, just add

```
module load nwchem/6.3_R2_itl
```

in your job script.

Job Script Example

```
#!/bin/sh
#PBS -N job
#PBS -l nodes=1:ppn=8
#PBS -l cput=10:00:00
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe

#=======================================================#
# USER CONFIG
#=======================================================#
INPUT_FILE="3carbo_dft.nw"
OUTPUT_FILE="3carbo_dft.out"
MODULE_NAME="nwchem/6.3_R2_itl"
PROGRAM_NAME="nwchem"
#=======================================================#
# MODULE is loaded
#=======================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=======================================================#
# SCRATCH directory is created at the local disks
#=======================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
```

```
fi
export SCRATCH_DIR=$SCRDIR

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $PBS_O_WORKDIR
# Run a MPI application
mpirun -np $NP $PROGRAM_NAME $INPUT_FILE >& $OUTPUT_FILE

#=========================================================#
# RESULTS are migrated back to the working directory
#=========================================================#
echo "COPYING SCRACH FILES TO " $PBS_O_WORKDIR
cp -rp $SCRDIR/* $PBS_O_WORKDIR
if [ $? != 0 ]; then
    {
        echo "Sync ERROR: problem copying files from $tdir to $PBS_O_WORKDIR;"
        echo "Contact HPC admin for a solution."
        exit 1
    }
fi

#=========================================================#
# DELETING the local scratch directory
#=========================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi

#=========================================================#
# ALL DONE
#=========================================================#
```

The example test can be found in `/hpc/tmp/examples/nwchem/6.3_R2/qmd_home`.

F.9.18. OpenBabel

Open Babel is a chemical toolbox designed to speak the many languages of chemical data. It's an open, collaborative project allowing anyone to search, convert, analyze, or store data from molecular modeling, chemistry, solid-state materials, biochemistry, or related areas.

Version History

- 2013/12/03 OpenBabel 2.3.2

How to Use

To use OpenBabel, load modules as below

```
module load openbabel/2.3.2
```

For more details on using OpenBabel, please read its online tutorials.

F.9.19. ORCA

ORCA is a flexible, efficient and easy-to-use general purpose tool for quantum chemistry with specific emphasis on spectroscopic properties of open-shell molecules. It features a wide variety of standard quantum chemical methods ranging from semiempirical methods to DFT to single- and multireference correlated *ab initio* methods. It can also treat environmental and relativistic effects.

Due to the user-friendly style, ORCA is considered to be a helpful tool not only for computational chemists, but also for chemists, physicists and biologists that are interested in developing the full information content of their experimental data with help of calculations.

Version History

- 2014/12/15 ORCA 3.0.3 PGI Build
- 2014/07/23 ORCA 3.0.2 PGI Build
- 2014/01/08 ORCA 3.0.1 PGI Build

How to Use

To use ORCA in the double precision add the following line in your job script:

```
module load orca/3.0.1_pgi
module load orca/3.0.2_pgi
module load orca/3.0.3_pgi
```

Job Script Example

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l cput=30:00:00
#PBS -j eo
#PBS -e test.err
#PBS -N orca.job

#=====================================================#
# USER CONFIG
#=====================================================#
# Input file name
INPUT_FILE=test.inp
# Output file name
OUTPUT_FILE=test.out
# Module name
MODULE_NAME=orca/3.0.3_pgi
# Program name
PROGRAM_NAME=orca

#=====================================================#
# MODULE is loaded
#=====================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME

#=====================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=====================================================#
cd $PBS_O_WORKDIR
```

```
$PROGRAM_NAME $INPUT_FILE >& $PBS_O_WORKDIR/$OUTPUT_FILE

#=====================================================#
# ALL DONE
#=====================================================#
```

The above example test can be found at

`/hpc/tmp/examples/orca/3.0.3_pgi`

Note: to run in parallel you need the `%pal` keyword in your input file. Make sure to request the same number of cores in the job script and the input file.

By using this software we suppose you have registered at the Orca website and accepted all license terms in it. Please register if you havent done it.

F.9.20. Q-Chem

Q-Chem is a comprehensive ab initio quantum chemistry package for accurate predictions of molecular structures, reactivities, and vibrational, electronic and NMR spectra. The new release of Q-Chem 4.0 represents the state-of-the-art of methodology from the highest performance DFT/HF calculations to high level post-HF correlation methods:

- Dispersion-corrected and double hybrid DFT functionals;
- Faster algorithms for DFT, HF and coupled-cluster calculations;
- Structures and vibrations of excited states with TD-DFT;
- Methods for mapping complicated potential energy surfaces;
- Efficient valence space models for strong correlation;
- More choices for excited states, solvation and charge-transfer;
- Effective Fragment Potential and QM/MM for large systems;
- Shared-memory for multicores and implementations for GPUs.

Version History

- 2015/01/08 Q-Chem 4.2.0
- 2014/01/20 Q-Chem 4.0.0.1

How to Use

To use Q-Chem, load one of the following modules

```
module load qchem/4.0.0.1
module load qchem/4.2.0
```

QChem is a commercial licensed software. Please contact HPC admin on how to access it.

Job Script Example

An example jobs script is shown below and can be found at
`/hpc/tmp/examples/qchem/4.2.0`

```
#!/bin/bash
#PBS -l nodes=1:ppn=8
#PBS -l mem=32GB
#PBS -l cput=1000:00:00
#PBS -j eo
#PBS -e test.err
#PBS -N qchem.job
#PBS -m ae

#=========================================================#
# USER CONFIG
#=========================================================#
# Input file name
INPUT_FILE=2bset_h2o.in
# Output file name
OUTPUT_FILE=2bset_h2o.out
# Module name
MODULE_NAME=qchem/4.2.0
# Program name
PROGRAM_NAME=qchem

#=========================================================#
# MODULE is loaded
#=========================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $PBS_O_WORKDIR
$PROGRAM_NAME -pbs -np $NP $INPUT_FILE >& $OUTPUT_FILE

#=========================================================#
# ALL DONE
#=========================================================#
```

For more details on using Quantum Espresso, please read its online tutorials.

F.9.21. Quantum ESPRESSO

Quantum ESPRESSO is an integrated suite of computer codes for electronic-structure calculations and materials modeling at the nanoscale. It is based on density-functional theory, plane waves, and pseudopotentials (both norm-conserving and ultrasoft).

Version History

- 2015/03/17 Quantum Espresso 5.1.2 PGI build
- 2014/11/21 Quantum Espresso 5.1.1 PGI build
- 2014/09/01 Quantum Espresso 5.1.0 PGI build
- 2013/11/01 Quantum Espresso 5.0.3 OpenMPI+PGI build

How to Use

To use Quantum ESPRESSO, load one of the following modules

```
module load quantum_espresso/5.1.2_pgi
module load quantum_espresso/5.1.1_pgi
```

```
module load quantum_espresso/5.1.0_pgi
module load quantum_espresso/5.0.3_openmpi_pgi
```

Job Script Example

An example jobs script is shown below and can be found at `/hpc/tmp/examples/quantum_espresso/5.1.2_pgi/runjob.sh`

```
#!/bin/sh
#PBS -N espresso.job
#PBS -l cput=01:00:00
#PBS -l pvmem=2GB
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe
#PBS -l nodes=1:ppn=8

#=========================================================#
# USER CONFIG
#=========================================================#
INPUT_FILE=electric0.in
OUTPUT_FILE=electric0.out
MODULE_NAME=quantum_espresso/5.1.2_pgi
UPF_MODULE=upf_files/sep102014
PROGRAM_NAME=pw.x
# do NOT copy the scratch files back
COPY_SCRATCH_BACK=false
#=========================================================#
# MODULE is loaded
#=========================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
module load $UPF_MODULE
cat $PBS_NODEFILE

#=========================================================#
# SCRATCH directory is created at the local disks
#=========================================================#
if grep -q outdir $PBS_O_WORKDIR/$INPUT_FILE
then
echo "Please remove the outdir flag in " $PBS_O_WORKDIR/$INPUT_FILE
exit 0
fi
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
fi
export ESPRESSO_TMPDIR=$SCRDIR

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $PBS_O_WORKDIR
mpirun -np $NP $PROGRAM_NAME < $INPUT_FILE >& $OUTPUT_FILE

#=========================================================#
# RESULTS are migrated back to the working directory
#=========================================================#
if [[ "$COPY_SCRATCH_BACK" == *true* ]]
then
    echo "COPYING SCRACH FILES TO " $PBS_O_WORKDIR/$PBS_JOBID
    cp -rp $SCRDIR $PBS_O_WORKDIR
    if [ $? != 0 ]; then
        {
            echo "Sync ERROR: problem copying files from $tdir to $PBS_O_WORKDIR;"
            echo "Contact HPC admin for a solution."
            exit 1
        }
```

```
    fi
fi

#=====================================================#
# DELETING the local scratch directory
#=====================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi
#=====================================================#
# ALL DONE
#=====================================================#
```

Note: You could use the your own pseudopotential files or the official ones by load the module of

```
upf_files/sep102014
```

For more details on using Quantum Espresso, please read its online tutorials.

F.9.22. SIESTA

SIESTA is both a method and its computer program implementation, to perform efficient electronic structure calculations and ab initio molecular dynamics simulations of molecules and solids. SIESTA's efficiency stems from the use of strictly localized basis sets and from the implementation of linear-scaling algorithms which can be applied to suitable systems. A very important feature of the code is that its accuracy and cost can be tuned in a wide range, from quick exploratory calculations to highly accurate simulations matching the quality of other approaches, such as plane-wave and all-electron methods.

Version History

- 2015/02/11 Siesta 3.2 p5 Intel compiler build
- 2014/02/11 Siesta 3.2 p4 Intel compiler build

How to Use

To use Siesta, load module as below

```
module load siesta/3.2_p5_itl
module load siesta/3.2_p4_itl
```

Job Script Example

An example jobs script is shown below and can be found as `/hpc/tmp/examples/siesta/3.2_p5/runjob.sh`

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l mem=16GB
#PBS -l cput=100:00:00
#PBS -j eo
#PBS -e job.std.out
```

```
#PBS -N siesta.job
#PBS -m ae

#=========================================================#
# USER CONFIG
#=========================================================#
# Input file name
INPUT_FILE=sih.fdf
# Output file name
OUTPUT_FILE=sih.log
# Module name
MODULE_NAME=siesta/3.2_p5_itl
# Program name
PROGRAM_NAME=siesta

#=========================================================#
# MODULE is loaded
#=========================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=========================================================#
# SCRATCH directory is created at the local disks
#=========================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
fi

#=========================================================#
# TRANSFER input files to the scratch directory
#=========================================================#
# copy everything (Option)
cp $PBS_O_WORKDIR/* $SCRDIR

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $SCRDIR
mpiexec $PROGRAM_NAME  < $INPUT_FILE >& $PBS_O_WORKDIR/$OUTPUT_FILE

#=========================================================#
# RESULTS are migrated back to the working directory
#=========================================================#
echo "COPYING SCRACH FILES TO " $PBS_O_WORKDIR
cp -rp $SCRDIR/* $PBS_O_WORKDIR
if [ $? != 0 ]; then
    {
        echo "Sync ERROR: problem copying files from $tdir to $PBS_O_WORKDIR;"
        echo "Contact HPC admin for a solution."
        exit 1
    }
fi

#=========================================================#
# DELETING the local scratch directory
#=========================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi
#=========================================================#
# ALL DONE
#=========================================================#
```

For more details on using Quantum Espresso, please read its online documents.

F.9.23. VMD

VMD is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting.

Version History

- 2015/01/07 VMD 1.9.2
- 2013/10/22 VMD 1.9.1

How to Use

To use VMD, load one of VMD module as below

```
module load vmd/1.9.1
```

or

```
module load vmd/1.9.2
```

To run VMD in the Graphic mode, first submit an interactive job to the queue system by

```
qsub -I -X -q short
```

Then load the 'VMD' module and run 'vmd' command to start.

```
module load vmd/1.9.2
vmd
```

Remember to type 'exit' to quit the interactive job.

F.9.24. WIEN2K

WIEN2K allows to perform electronic structure calculations of solids using density functional theory (DFT). It is based on the full-potential (linearized) augmented plane-wave ((L)APW) + local orbitals (lo) method, one among the most accurate schemes for band structure calculations. WIEN2k is an all-electron scheme including relativistic effects and has many features. It has been licensed by more than 2000 user groups.

Version History

- 2013/11/14 WIEN 2K 12.1 Intel Compiler Build

How to Use

To use WIEN2K, load module as below

```
module load wien2k/12.1_itl_mkl
```

WIEN2k deployed on the cluster is licensed by the specific group. Please contact IMTS for details.

F.9.25. XCrySDen

XCrySDen is a crystalline and molecular structure visualisation program, which aims at display of isosurfaces and contours, which can be superimposed on crystalline structures and interactively rotated and manipulated. It can run on most UNIX platforms, without any special hardware requirements.

Version History

- 2014/10/18 XCrySDen 1.5.60
- 2013/11/11 XCrySDen 1.5.53

How to Use

To use XCrySDen, please submit an interactive job to 'short' queue and then load one of the 'xcrysden' modules, i.e.

```
-sh-4.1$ qsub -I -X -q short
-sh-4.1$ module load xcrysden/1.5.53
-sh-4.1$ xcrysden
```

or

```
-sh-4.1$ qsub -I -X -q short
-sh-4.1$ module load xcrysden/1.5.60
-sh-4.1$ xcrysden
```

Remember typing 'exit' after completing all works to quit the interactive job.

F.10. Informatics

F.10.1. Caffe

Caffe is a framework for convolutional neural network algorithms, developed with speed in mind. Caffe aims to provide computer vision scientists and practitioners with a clean and modifiable implementation of state-of-the-art deep learning algorithms. For example, network structure is easily specified in separate config files, with no mess of hard-coded parameters in the code.
At the same time, Caffe fits industry needs, with blazing fast C++CUDA code for GPU computation. Caffe is currently the fastest GPU CNN implementation publicly available, and is able to process more than 40 million images per day with a single NVIDIA K40 or Titan GPU (or 20 million images per day on a K20 GPU). Thats 192 images per second during training and 500 images per second during test.
Caffe also provides seamless switching between CPU and GPU, which allows one to train models with fast GPUs and then deploy them on non-GPU clusters with one line of code: `Caffe::set_mode(Caffe::CPU)`. Even in CPU mode, computing predictions on an image takes only 20 ms when images are processed in batch mode. While in GPU mode, computing predictions on an image takes only 2 ms when images are processed in batch mode.

Version History

- 2015/02/16 Caffe 20150216 MKL build
- 2014/06/16 Caffe 20140616 MKL build

How to Use

Load one of the following caffe modules as below:

```
module load caffe/20150216
```

or

```
module load caffe/20140616
```

Job Script Example

An example job script is given below and can be found in the MNIST example case under `/hpc/tmp/examples/caffe/mnist`:

```
#!/bin/sh
#PBS -N job
#PBS -l nodes=1:ppn=1
#PBS -l cput=100:00:00
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe

#=====================================================#
# USER CONFIG
#=====================================================#
INPUT_FILE="lenet_solver.prototxt"
OUTPUT_FILE="lenet_solver.log"
MODULE_NAME="caffe/20140616_mkl"
PROGRAM_NAME="train_net.bin"

#=====================================================#
# MODULE is loaded
#=====================================================#
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE
NP=`wc -l < $PBS_NODEFILE`
export OMP_NUM_THREADS=$NP

#=====================================================#
# PROGRAM is executed with the output or log file #
# direct to the working directory #
#=====================================================#
cd $PBS_O_WORKDIR
# Run a system wide sequential application
GLOG_logtostderr=1 $PROGRAM_NAME $INPUT_FILE >& $OUTPUT_FILE

#=====================================================#
# ALL DONE
#=====================================================#
```

For more details on using Caffe, please read its online documents.

F.10.2. netCDF

NetCDF (network Common Data Form) is a set of interfaces for array-oriented data access and a freely distributed collection of data access libraries for C, Fortran, C++,

Java, and other languages. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data.

NetCDF data is:

- Self-Describing. A netCDF file includes information about the data it contains.
- Portable. A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- Scalable. A small subset of a large dataset may be accessed efficiently.
- Appendable. Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- Sharable. One writer and multiple readers may simultaneously access the same netCDF file.
- Archivable. Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

Version History

- 2014/02/11 netCDF Fortran library 4.2 Intel compiler build
- 2014/02/11 netCDF C library 4.3.0 Intel compiler build
- 2014/01/21 netCDF Fortran library 4.2 GCC build
- 2014/01/21 netCDF C++ library 4.2.1 GCC build
- 2013/11/05 netCDF C library 4.3.0 GCC build

How to Use

The netCDF library is installed at

```
/hpc/software/package/netcdf/4.3.0/lib
/hpc/software/package/netcdf/4.3.0_itl/lib
```

The netCDF C++ library is installed at

```
/hpc/software/package/netcdf-cxx/4.2.1/lib
```

The netCDF Fortran library is installed at

```
/hpc/software/package/netcdf-fortran/4.2/lib
/hpc/software/package/netcdf-fortran/4.2_itl/lib
```

For more details on using netCDF, please read its online documents.

F.10.3. netCDF Operator (NCO)

netCDF Operator (NCO) manipulates data stored in netCDF format. It also exploits the geophysical expressivity of many CF (Climate & Forecast) metadata conventions, the flexible description of physical dimensions translated by UDUnits, the network transparency of OPeNDAP, the storage features (e.g., compression, chunking, groups) of HDF (the Hierarchical Data Format), and many powerful mathematical and statistical algorithms of GSL (the GNU Scientific Library). NCO is fast, powerful, and free.

Version History

- 2013/11/05 NCO 4.3.7

How to Use

p To use NCO, load the module as below

```
module load nco/4.3.7
```

For more details on using NCO, please read its online documents.

F.10.4. RepastHPC

Repast for High Performance Computing(Repast HPC) is a next generation agent-based modeling system intended for large-scale distributed computing platforms. It implements the core Repast Simphony concepts (e.g. contexts and projections), modifying them to work in a parallel distributed environment. Repast HPC is written in cross-platform C++. It can be used on workstations, clusters, and supercomputers running Apple Mac OS X, Linux, or Unix. Portable models can be written in either standard or Logo-style C++.

Version History

- 2013/11/05 RepastHPC 2.0

How to Use

To use RepastHPC, add

```
module load repasthpc/2.0
```

in your job script. For more details on using RepastHPC, please read its online documents.

F.10.5. SUMO(Simulation of Urban Mobility)

SUMO is an open source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center. SUMO is open source, licensed under the GPL.

Version History

- 2013/11/14 SUMO 0.18.0

How to Use

To use AutoDock Vina, adding

```
module load sumo/0.18.0
```

in your job script. For more details on using SUMO, please read its online tutorials.

F.11. Engineering

F.11.1. MATLAB

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation.

MATLAB installed at the HPC cluster support a wide range of toolboxes as listed below:

```
MATLAB
Simulink
Bioinformatics Toolbox
Communications System Toolbox
Computer Vision System Toolbox
Control System Toolbox
Curve Fitting Toolbox
DSP System Toolbox
Database Toolbox
Econometrics Toolbox
Embedded Coder
Financial Instruments Toolbox
Financial Toolbox
Fixed-Point Designer
Fuzzy Logic Toolbox
Global Optimization Toolbox
HDL Coder
Image Acquisition Toolbox
Image Processing Toolbox
Instrument Control Toolbox
MATLAB Coder
MATLAB Compiler
MATLAB Report Generator
Mapping Toolbox
Neural Network Toolbox
Optimization Toolbox
Parallel Computing Toolbox
Partial Differential Equation Toolbox
RF Toolbox
Robust Control Toolbox
Signal Processing Toolbox
SimBiology
SimElectronics
SimEvents
SimHydraulics
SimMechanics
SimPowerSystems
SimRF
Simscape
Simulink 3D Animation
Simulink Coder
Simulink Control Design
Simulink Design Optimization
Simulink Report Generator
Stateflow
Statistics Toolbox
Symbolic Math Toolbox
System Identification Toolbox
SystemTest
Wavelet Toolbox
```

Version History

- 2014/11/17 MATLAB R2014b
- 2014/03/27 MATLAB R2014a
- 2013/11/26 MATLAB R2013b
- 2013/11/05 MATLAB R2011a
- 2013/10/21 MATLAB R2013a

How to Use

To use MATLAB, add the corresponding module in your job script:

```
module load matlab/r2011a
module load matlab/r2013a
module load matlab/r2013b
module load matlab/r2014a
module load matlab/r2014b
```

NOTE: Always use the `-singleCompThread` flag when start your MATLAB jobs.

Job Script Example

Example job scripts can be found under `/hpc/tmp/examples/matlab`:

```
-sh-4.1$ more runjob.sh
#!/bin/sh
#PBS -N test
#PBS -l cput=48:00:00
#PBS -l mem=1GB
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe
#PBS -l nodes=1:ppn=1

#===============INFORMATION==========================#
# This is the UOW HPC cluster job script template for
# running MATLAB jobs at the local disks.  Fill in the
# USER CONFIG portion and submit the job to the queue
# using the qsub command.  In some cases, you may need
# to revise lines within the PROGRAM portion.
#====================================================#

#====================================================#
# USER CONFIG
#====================================================#
INPUT_FILE="md_pool.m"
OUTPUT_FILE="output"
MODULE_NAME="matlab/r2014b"
PROGRAM_NAME="matlab"
# Set as true if you need those scratch files.
COPY_SCRATCH_BACK=
#====================================================#
# MODULE is loaded
#====================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#====================================================#
# SCRATCH directory is created at the local disks
#====================================================#
SCRDIR=/export/scratch/common/$PBS_JOBID
if [ ! -d "$SCRDIR" ]; then
mkdir $SCRDIR
fi

#====================================================#
# TRANSFER input files to the scratch directory
#====================================================#
# just copy input file
#cp $PBS_O_WORKDIR/$INPUT_FILE $SCRDIR
# copy everything (Option)
cp $PBS_O_WORKDIR/* $SCRDIR

#====================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#====================================================#
echo "START TO RUN WORK"
cd $SCRDIR
export NWORKERS=$NP
$PROGRAM_NAME -singleCompThread -nosplash -nodesktop $INPUT_FILE  > $PBS_O_WORKDIR/$OUTPUT_FILE

#====================================================#
# RESULTS are migrated back to the working directory
#====================================================#
if [[ "$COPY_SCRATCH_BACK" == *true* ]]
then
    echo "COPYING SCRACH FILES TO " $PBS_O_WORKDIR/$PBS_JOBID
    cp -rp $SCRDIR/* $PBS_O_WORKDIR
```

```
    if [ $? != 0 ]; then
        {
            echo "Sync ERROR: problem copying files from $tdir to $PBS_O_WORKDIR;"
            echo "Contact HPC admin for a solution."
            exit 1
        }
    fi
fi

#=========================================================#
# DELETING the local scratch directory
#=========================================================#
cd $PBS_O_WORKDIR
if [[ "$SCRDIR" == *scratch* ]]
then
    echo "DELETING SCRATCH DIRECTORY" $SCRDIR
    rm -rf $SCRDIR
    echo "ALL DONE!"
fi
#=========================================================#
# ALL DONE
#=========================================================#
```

Unless you are using Parallel Computing Toolbox (PCT), please always request 1 core in your MATLAB jobs.

F.11.2. ANSYS,FLUENT,LSDYNA,ANSYS Electromagnetics Suite

ANSYS Mechanical and ANSYS Multiphysics software are non exportable analysis tools incorporating pre-processing (geometry creation, meshing), solver and post-processing modules in a graphical user interface. These are general-purpose finite element modeling packages for numerically solving mechanical problems, including static/dynamic structural analysis (both linear and non-linear), heat transfer and fluid problems, as well as acoustic and electro-magnetic problems.

Version History

- 2015/03/31 ANSYS Electromagnetics Suite 16.0
- 2013/11/19 ANSYS 14.5
- 2013/11/19 LS-DYNA 14.5

How to Use

Load one of the following modules in your job script:

```
module load ansysem/16.0
module load ansys/14.5
module load lsdyna/14.5
```

ANSYS is a commercial licensed software. Please contact HPC admin to learn the details.

F.11.3. ABAQUS

ABAQUS suite of engineering analysis software packages is used to simulate the physical response of structures and solid bodies to load, temperature, contact,impact, and other environmental conditions.

Version History

- 2013/12/14 ABAQUS 6.9-1
- 2013/11/29 ABAQUS 6.12-1

How to Use

ABAQUS is a commercial licensed software. Please contact HPC admin to learn the details.

F.11.4. LAMMPS

LAMMPS stands for Large-scale Atomic/Molecular Massively Parallel Simulator. LAMMPS has potentials for soft materials (biomolecules, polymers) and solid-state materials (metals, semiconductors) and coarse-grain systems. It can be used to model atoms or, more generically, as a parallel particle simulator at the mesoscale or continuum levels.

Version History

- 2014/11/27 LAMMPS 30Oct14
- 2014/01/07 LAMMPS 5Nov10
- 2013/12/03 LAMMPS 1DEC13

How to Use

To use LAMMPS, add one of the following modules in the job script.

```
module load lammps/30Oct14
module load lammps/1Dec13
module load lammps/5Nov10
```

Job Script Example

An example job script is shown as below and can be found in `/hpc/tmp/examples/lammps/1Dec13/crack`

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l cput=30:00:00
#PBS -j eo
#PBS -e test.err
#PBS -N lmk.job

#=====================================================#
```

```
# USER CONFIG
#=========================================================#
# Input file name
INPUT_FILE=in.crack
# Output file name
OUTPUT_FILE=test.log
# Module name
MODULE_NAME=lammps/1Dec13
# Program name
PROGRAM_NAME=lmp
#=========================================================#
# MODULE is loaded
#=========================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $PBS_O_WORKDIR
mpirun -np $NP $PROGRAM_NAME < $INPUT_FILE >& $OUTPUT_FILE

#=========================================================#
# ALL DONE
#=========================================================#
```

For more details on using LAMMPS, please read it's online tutorials.

F.11.5. Materials Studio

Materials Studio is a comprehensive materials modeling and simulation application designed for scientists in chemicals and materials R&D as well as pharmaceuticals development. At present, we have 3 modules installed on the cluster: amorphouscell, discover and compass.

Version History

- 2013/12/10 Materials Studio 7.0

How to Use

Materials Studio is a commercial licensed software. Please contact HPC admin to get the access to it.

F.12. Biology

F.12.1. ATSAS

ATSAS is a program suite for small-angle scattering data analysis from biological macromolecules.

Version History

- 2013/11/06 ATSAS 2.5.1-1

How to Use

To use ATSAS, add one of the following modules in the job script.

```
module load atsas/2.5.1-1
```

For more details on using ATSAS, please read its online manual.

F.12.2. MrBayes

MrBayes is a program for the Bayesian estimation of phylogeny. Bayesian inference of phylogeny is based upon a quantity called the posterior probability distribution of trees, which is the probability of a tree conditioned on the observations. The conditioning is accomplished using Bayes's theorem. The posterior probability distribution of trees is impossible to calculate analytically; instead, MrBayes uses a simulation technique called Markov chain Monte Carlo (or MCMC) to approximate the posterior probabilities of trees.

Since version 3.2, MrBayes prints all parameter values of all chains (cold and heated) to a checkpoint file every `Checkfreq` generations, by default every 100,000 generations. The checkpoint file has the suffix `.ckp`. If you run an analysis and it is stopped prematurely, you can restart it from the last checkpoint by using `mcmc append=yes`.MrBayes will start the new analysis from the checkpoint; it will even read in all the old trees and include them in the convergence diagnostics. At the end of the new run, you will have parameter and tree files that are indistinguishable from those you would have obtained from an uninterrupted analysis. Our data set is so small, however, that we are likely to get an adequate sample from the posterior before the first checkpoint is reached.

Version History

- 2013/10/24 Mrbayes 3.2.2

How to Use

To use Mrbayes, add one of the following modules in the job script.

```
module load mrbayes/3.2.2
```

Job Script Example

An example job script is shown as below and can be found in `/hpc/tmp/examples/mrbayes/3.2.2`

```
#!/bin/sh
#PBS -N mrbayes
#PBS -l cput=01:00:00
#PBS -l mem=1GB
#PBS -j eo
#PBS -e test.err
#PBS -m abe
#PBS -l nodes=1:ppn=8

#=======================================================#
# USER CONFIG
#=======================================================#
# Input file name
```

```
INPUT_FILE=hymfossil.nex
# Output file name
OUTPUT_FILE=hymfossil.log
# Module name
MODULE_NAME=mrbayes/3.2.2
# Program name
PROGRAM_NAME=mb

#=========================================================#
# MODULE is loaded
#=========================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=========================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=========================================================#
cd $PBS_O_WORKDIR
mpirun -np $NP $PROGRAM_NAME $INPUT_FILE >& $OUTPUT_FILE

#=========================================================#
# ALL DONE
#=========================================================#
```

For more details on using Mrbayes, please read it's online manual.

F.12.3. PartitionFinder

PartitionFinder is free open source software to select best-fit partitioning schemes and models of molecular evolution for phylogenetic analyses. Most of the lab's coding output currently goes into PartitionFinder.

Version History

- 2014/05/20 PartitionFinder 1.1.1

How to Use

To use PartitionFinder, add the following modules in the job script.

```
module load partitionfinder/1.1.1
```

Job Script Example

An example job script is shown as below and can be found under `/hpc/tmp/examples/partitionfinder/1.1.1/nucleotide`.

```
#!/bin/sh
#PBS -N job
#PBS -l nodes=1:ppn=4
#PBS -l cput=10:00:00
#PBS -j eo
#PBS -e job.std.out
#PBS -m abe

#=========================================================#
# USER CONFIG
#=========================================================#
```

```
INPUT_DIR="/hpc/tmp/examples/partitionfinder/1.1.1/nucleotide"
OUTPUT_FILE="logfile"
MODULE_NAME="partitionfinder/1.1.1"
PROGRAM_NAME="PartitionFinder.py"
#Option:  see program manual for additional flags.
#FLAGS="--raxml"
#=======================================================#
# MODULE is loaded
#=======================================================#
NP=`wc -l < $PBS_NODEFILE`
source /etc/profile.d/modules.sh
module load $MODULE_NAME
cat $PBS_NODEFILE

#=======================================================#
# PROGRAM is executed with the output or log file
# direct to the working directory
#=======================================================#
cd $PBS_O_WORKDIR
# Run a system wide sequential application
$PROGRAM_NAME -p $NP $FLAGS $INPUT_DIR >& $OUTPUT_FILE

#=======================================================#
# ALL DONE
#=======================================================#
```

Another example of running `PartitionFinderProtein.py` can be found under `/hpc/tmp/examples/partitionfinder/1.1.1/aminoacid`
To run your own job, please specify the input directory (absolute path) and output files in the line of '`INPUT_DIR`' and '`OUTPUT_FILE`'. Only two programs available for the '`PROGRAM_NAME`' line, i.e.

- `PartitionFinder.py`
- `PartitionFinderProtein.py`

You could also add additional flags like '`--raxml`' in the '`FLAGS`' line and remove the heading '`#`'. Please check the program manual for other flags you can use.

For more details on using PartitionFinder, please read the online documents.

F.12.4. QIIME

QIIME (canonically pronounced "chime") stands for Quantitative Insights Into Microbial Ecology. QIIME is an open source software package for comparison and analysis of microbial communities, primarily based on high-throughput amplicon sequencing data (such as SSU rRNA) generated on a variety of platforms, but also supporting analysis of other types of data (such as shotgun metagenomic data). QIIME takes users from their raw sequencing output through initial analyses such as OTU picking, taxonomic assignment, and construction of phylogenetic trees from representative sequences of OTUs, and through downstream statistical analysis, visualization, and production of publication-quality graphics. QIIME has been applied to studies based on billions of sequences from thousands of samples.

Version History

- 2014/05/05 QIIME 1.8.0
- 2013/11/13 QIIME 1.7.0

How to Use

To use Mrbayes, add one of the following modules in the job script.

```
module load qiime/1.7.0
```

or

```
module load qiime/1.8.0
```

For more details on using QIIME, please read it's online documents.